

Model Selection for Latent Force Models



Cristian Guarnizo Lemus

Universidad Tecnológica de Pereira

This dissertation is submitted for the degree of
Doctor of Engineering

Advisor

Dr. Mauricio A. Álvarez López.

Committee

Dr. Carl Henrik Ek.

Dr. Julián D. Arias Londoño.

Dr. Andrés M. Álvarez Meza.

Dr. Julián D. Echeverry Correa.

Date

May 30, 2018.

to the warrior princess Alana.

Acknowledgements

I would firstly like to thank my supervisor Dr. Mauricio A. Álvarez for his support, inspiration and guidance during these years. Mauricio was always kind and available to discuss every little detail of my research work. Additionally, he trusted me with so many interesting ideas and encourage me to pursuit my own ideas.

Thanks to all friends and colleagues at the Automatics research group. It was a privilege the get to know each one of them. Especially, I would like to thank Carlos D. Zuluaga, Jhouben J. Cuesta, Andres F. López, Juan F. López, Hernan F. Garcia, Pablo A. Alvarado, Hernan D. Vargas, Diego A. Agudelo, Julian Gil and Cristian A. Torres for their friendship and support during this process. Besides, I am very grateful to Dr. Neil D. Lawrence and Dr. Simo Särkkä for allowing me to have the chance to spend highly valuable time at their research labs.

I would like to thank my thesis defence committee: Dr. Carl Henrik Ek, Dr. Julian D. Arias, Dr. Andres M. Álvarez, and Dr. Julian D. Echeverry for their comments and suggestions that allow me to improve this document.

I would like to acknowledge my source of funding given by Convocatoria 567 from Administrative Department of Science, Technology and Innovation of Colombia (COL-CIENCIAS).

Finally, I want to thank my family for their love and support. Especially, thanks to Anatol Guarnizo and Marlene S. Lemus for their never-ending love and patience.

Abstract

Many engineering and science processes can be described by ordinary differential equations (ODEs), for example, the level of protein regulated by a transcription factor in gene expression data can be modelled by a first order ODE, or the displacement of a mass induced by the force applied through a system of spring and damper elements is modelled using a second order ODE. In some tasks involving ODEs, it is of main interest to recover the forcing function (e.g. the protein in gene expression data, and the force in a mechanical system). The forcing function can be estimated from output data by knowing before hand the ODE's order and how it is parametrized. Taking into account the above and placing a Gaussian Process (GP) prior on the forcing function, it is possible to learn system's parameters and estimate the forcing function by using latent force models (LFMs) (Álvarez et al., 2013). LFMs are hybrid models that combine a mechanistic model with a data driven model. This is done by encoding the ODE's information within the covariance function, by means of the convolution between the input function (which is modelled with a GP prior) and the system's impulse response function. Even though LFMs have been considered as a promising approach to do extrapolation with Gaussian Processes, they assume that the number of latent forces and the impulse response functions are known.

In this thesis we explore several extensions of these models to address these limitations. In the first proposed method, the number of latent forces (forcing functions) is automatically selected by means of the non-parametric Indian Buffet Process (IBP) prior. Additionally, the IBP allows us to estimate the sparse interconnection between the outputs and the latent forces.

Moving on the next topic, we estimate the impulse response function (IRF) of linear time-invariant systems using Laguerre functions using LFMs and Sequential LFMs. Those approaches are tested on multiple-input multiple-output systems and missing data scenarios.

Lastly, this thesis additionally develops methods focused on the estimation of the latent forces and IRFs over non-linear dynamical system cases known as Wiener systems.

Contents

List of Figures	ix
List of Tables	xi
Notation	xii
1 Introduction	1
1.1 Aims	2
1.1.1 General aim	2
1.1.2 Specific aims	2
1.2 Outline and contributions	2
1.2.1 Number of latent forces	2
1.2.2 Modelling multiple-input multiple-output data	4
1.2.3 Wiener systems and latent force models	4
1.3 Software and publications	5
2 Linear latent force models	6
2.1 Background	6
2.1.1 Linear dynamical systems	6
2.1.2 Gaussian Processes	8
2.1.3 Gaussian Processes and Linear Operators	10
2.2 Latent force models	11
2.2.1 Model definition	12
2.2.2 First order ordinary differential equation (ODE1)	14
2.2.3 Second order ordinary differential equation (ODE2)	14
2.2.4 Gaussian smoothing (GS)	15
2.2.5 Inference	16
2.2.6 Predictive distributions	16

2.3	Sequential Latent Force models	17
2.3.1	Sequential Gaussian Process	17
2.3.2	Model definition	18
2.3.3	Sequential inference	18
2.3.4	Predictive distributions	21
2.4	Comparison of Latent Force model approaches	21
2.4.1	Discrete time or Continuous time	21
2.4.2	Computational complexity	21
2.4.3	Non-linear dynamic systems	22
3	Automatic selection of the number of latent forces	23
3.1	Indian Buffet Process	24
3.2	Model definition	24
3.3	Variational Inference approach	25
3.3.1	Updates for $q(v_q)$	28
3.3.2	Updates for $q(Z_{d,q})$	28
3.3.3	Posterior distribution for the latent forces	29
3.3.4	Hyperparameter learning	30
3.4	Predictive distribution	31
3.5	Related work	32
3.6	Results	33
3.6.1	Synthetic data	33
3.6.2	Yeast metabolic cycle data	36
3.6.3	Human motion capture data	38
3.6.4	Air temperature data	41
3.7	Discussion	42
3.8	Conclusions	44
4	Modelling multiple-input multiple-output data using Latent force models	45
4.1	Laguerre functions	46
4.2	Convolved Laguerre Process	46
4.2.1	Hyperparameter learning	47
4.2.2	Predictive distribution	48
4.3	Sequential Laguerre Processes	48
4.3.1	Sequential inference	50

4.3.2	Predictive distributions	50
4.4	Related work	50
4.5	Results	51
4.5.1	Approximation of the impulse response	51
4.5.2	Prediction of missing input/output values	52
4.5.3	CD-player arm	54
4.6	Conclusions	57
5	Wiener system approximation using latent force models	58
5.1	Linearisation Methods	58
5.2	Latent force models for Wiener systems	60
5.2.1	Inference	61
5.2.2	Predictive distribution	63
5.2.3	Related work	63
5.2.4	Experiments	64
5.3	Wiener system estimation based on sequential Laguerre processes	65
5.3.1	Inference	67
5.3.2	Experiments	68
5.4	Conclusions	70
6	Conclusions and Future Work	72
6.1	Conclusions	72
6.2	Future Work	73
	Appendix A Performance metrics	75
	Appendix B Extension for the estimation of the number of latent forces	76
B.1	Lower Bound terms description	76
B.2	Predictive distribution for latent forces	77
	References	78

List of Figures

1.1	Block diagram representation of the convolution process.	1
1.2	Diagram representation of a multiple-output latent force model explained by three latent forces.	3
1.3	Diagram representation of the Indian Buffet process and Latent force models.	3
1.4	Wiener system representation.	4
2.1	Block diagram of a LTI system.	7
3.1	Hinton diagrams for Toy experiment.	35
3.2	Estimation of the latent forces for experiment 3.6.1	35
3.3	Outputs prediction for the Toy experiment.	36
3.4	Hinton diagrams for the gene expression experiment.	37
3.5	Comparison of estimated latent forces for the toy experiment.	38
3.6	Hinton diagrams obtained assuming fixed full connectivity and using the proposed variational approach.	39
3.7	Prediction plots of testing values for experiment 3.6.3.	40
3.8	Hinton diagrams for Weather experiment.	42
4.1	Mean and two standard deviations calculated from the impulse response functions estimated using CLP and SLP for experiment 4.5.1.	52
4.2	Prediction of missing data (Test data) for the MIMO system described in section 4.5.2	53
4.3	Impulse response approximation for the MIMO system described in section 4.5.2, using the proposed methods CLP and SLP.	54
4.4	Prediction of missing data (Test data) for the CD-player arm system described in section 4.5.3	56

4.5	Impulse response approximation for the CD-player arm system described in section 4.5.3, using the proposed methods CLP and SLP.	57
5.1	Block representation of a Wiener system.	58
5.2	Plot of the forcing functions $u(t)$, true function, and the mean and two times the standard deviation of LFMs predictions for each Wiener system considered in experiments 5.2.4.	66
5.3	Comparison between the true IRF, and the mean and 2 times standard deviation of IRFs estimated by the proposed approach.	70
6.1	Block representation of a mutiple-input single-output system.	74

List of Tables

3.1	Description of the number of data points used for training and validation for the toy experiment 3.6.1	34
3.2	NMSE and NLPD measurements for testing data in toy experiment. . . .	36
3.3	Description of yeast data used in experiment 3.6.2	37
3.4	Description of MOCAP angles data used in experiment 3.6.3	39
3.5	NMSE and NLPD measurements for testing data in MOCAP experiment.	40
3.6	Description of Weather data used in experiment 3.6.4	41
3.7	Comparison of IBPLFM and CoGP methods based on NMSE and NLPD measurements for testing data on the weather experiment.	42
4.1	Comparison of CLP and SLP methods based on NMSE and NLPD measurements for the testing data of experiment 4.5.2	54
4.2	Description of CD-player arm data used in experiment 4.5.3	55
4.3	Comparison of CLP and SLP methods based on NMSE and NLPD measurements for the testing data of CD-player arm experiment.	55
5.1	Comparison of the Statistical and Taylor series linearisation methods based on NMSE and NLPD performance measurements for the prediction of the response function $f(t)$ at testing data.	65
5.2	NMSE from the 10 IRFs learned for experiment 5.3.2	69

Notation

Mathematical notation

Generalities

P	order of the ordinary differential equation
D	number of outputs
Q	number of inputs
T	sampling period for discrete models
N	total number of data points
N_d	number of data points for the d -th output
\mathbb{Z}	set of the integer numbers $\{1, 2, \dots, P\}$
t	input time

Operators

$\mathbb{E}[\cdot]$	expected value
$\text{tr}(\cdot)$	trace of a matrix
$H(q(x))$	Shannon entropy of $q(x)$
$\mathbf{A} \odot \mathbf{B}$	Hadamard product between matrices \mathbf{A} and \mathbf{B}

Functions

$G(t)$	Green's function or impulse response function
$u_q(t)$	q -th input, forcing or excitation function evaluated at t
$k_{u_q, u_q}(t, t')$	covariance function for the Gaussian process of $u_q(t)$
$f_d(t)$	d -th output or response function evaluated at t
$\mathbf{f}(t)$	vector-valued function, $\mathbf{f}(t) = [f_1(t), \dots, f_D(t)]^\top$
$k_{f_d, u_q}(t, t')$	cross-covariance between output $f_d(t)$ and function $u_q(t)$
$k_{f_d, f_{d'}}(t, t')$	cross-covariance between outputs $f_d(t)$ and $f_{d'}(t')$

Vectors and matrices

\mathbf{u}_q	$u_q(t)$ evaluated at \mathbf{t} or $\boldsymbol{\lambda}$, $\mathbf{u}_q = [u_q(\lambda_1), \dots, u_q(\lambda_M)]^\top$
\mathbf{u}	vectors $\{\mathbf{u}_q\}_{q=1}^Q$, stacked in one column vector
$\mathbf{K}_{\mathbf{u}_q, \mathbf{u}_q}$	covariance matrix with entries $k_{u_q, u_q}(t, t')$ evaluated at \mathbf{t}
$\mathbf{K}_{\mathbf{u}, \mathbf{u}}$	block-diagonal covariance matrix with blocks $\mathbf{K}_{\mathbf{u}_q, \mathbf{u}_q}$
\mathbf{t}_d	input time vector for output d , $\mathbf{t}_d = [t_{d,1}, \dots, t_{d,N_d}]^\top$
\mathbf{t}	vectors $\{\mathbf{t}_d\}_{d=1}^D$, stacked in one column vector
\mathbf{f}_d	$f_d(t)$ evaluated at \mathbf{t}_d , $\mathbf{f}_d = [f_d(t_{d,1}), \dots, f_d(t_{d,N_d})]^\top$
\mathbf{f}	vectors $\{\mathbf{f}_d\}_{d=1}^D$, stacked in one column vector
$\mathbf{K}_{\mathbf{f}_d, \mathbf{f}_{d'}}$	covariance matrix with entries $k_{f_d, f_{d'}}(t, t')$
$\mathbf{K}_{\mathbf{f}, \mathbf{f}}$	covariance matrix with block $\mathbf{K}_{\mathbf{f}_d, \mathbf{f}_{d'}}$
\mathbf{I}_N	identity matrix of size N
\mathbf{x}_k	state vector evaluated at time kT

Abbreviations

ODE	Ordinary Differential Equation
LTI	Linear Time-Invariant
GP	Gaussian Process
CGP	Convolved Gaussian Process
LFM	Latent Force Model
SLFM	Sequential Latent Force Model
IBP	Indian Buffet Process
CLP	Convolved Laguerre Process
SLP	Sequential Laguerre Process
NMSE	normalised mean square error
NLPD	negative log predictive density

Chapter 1

Introduction

Many engineering processes can be described by ordinary differential equations (ODEs), for example, the level of protein regulated by a transcription factor in gene expression data can be modelled by a first order ODE, or the displacement of a mass given by the force applied through a system of spring and damper elements is modelled using a second order ODE. In some tasks involving ODEs, it is of main interest to recover the forcing function (e.g. the protein in gene expression data, and the force applied on a mechanical system). The forcing function can be estimated from output data by knowing before hand the ODE's order and how it is parametrized. Taking into account the above and placing a Gaussian Process (GP) prior over the forcing function, it is possible to learn system's parameters and estimate the forcing function by using latent force models (LFMs) (Álvarez et al., 2013). LFMs are hybrid models that combine a mechanistic model with a data driven model. This is done by encoding the ODE's information within the covariance function, by means of the convolution between the forcing function (modelled by a GP prior) and the system's impulse response. Figure 1.1 depicts the convolution process, where the response function, $f(t)$, is obtained from the convolution of the excitation ($u(t)$) and the impulse response ($G(t)$) functions. Note

$$f(t) = \int G(t - \tau)u(\tau) d\tau$$
$$u(t) \longrightarrow \boxed{G(t)} \longrightarrow f(t)$$

Figure 1.1: Block diagram representation of the convolution process.

that the excitation function can be referred as forcing or input function in the context of LFMs.

As in [Rasmussen and Williams \(2006, Chapter 5\)](#), a broadly interpretation of model selection is adopted in this thesis. Model selection includes the choice of model parameters as well as the values for covariance function hyperparameters. In this thesis, we are interested in addressing several problems focused on dynamical systems using LFMs.

1.1 Aims

1.1.1 General aim

To develop probabilistic approaches to perform model selection on the number of latent functions and non-linear dynamics in latent force models.

1.1.2 Specific aims

1. To formulate probabilistic models which extend the Latent Force framework by automatically learning the number of latent functions.
2. To design inference methods for learning the number of latent functions.
3. To propose probabilistic models that extend the latent force model to represent non-linear dynamical systems.
4. To develop inference methods for recovering the forcing function in Non-linear dynamical systems.

1.2 Outline and contributions

The main contributions are briefly introduced in the following sections.

1.2.1 Number of latent forces

One of the main properties of LFMs is that they are able to straightforwardly explain multiple-output data using a set of shared latent forces. [Figure 1.2](#) shows a typical set-up of four outputs explained by three latent forces. Note that the set of shared latent forces induce dependency among the outputs. This dependency is useful to improve the predictions. However, despite LFMs' success for prediction, it is still unclear how to select the number of the latent forces ([Álvarez et al., 2012](#)), i.e. the number of shared

latent functions must be set by the experimenter or it can be found by testing the model with different number of latent functions and selecting the one that maximizes an objective function. In order to automatically infer the number of latent functions,

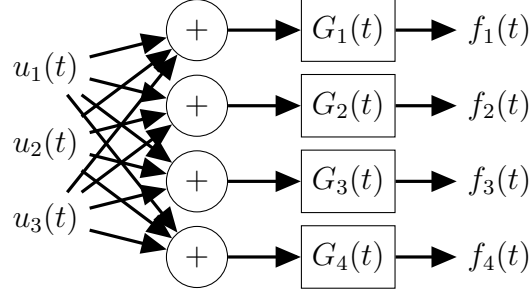


Figure 1.2: Diagram representation of a multiple-output latent force models explained by three latent forces.

we extend the LFM framework by controlling the number of latent forces and their relationship with the outputs by means of the Indian Buffet Process (IBP) prior. The IBP is a non-parametric prior over binary matrices, that imposes a structure over the sparsity pattern of the binary matrix (Griffiths and Ghahramani, 2005, 2011). Figure

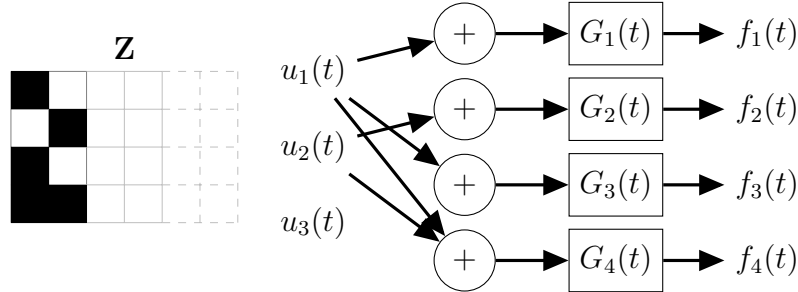


Figure 1.3: Diagram representation of the Indian Buffet process and Latent force models.

1.3 shows the binary matrix \mathbf{Z} obtained by sampling from the IBP prior. Note that the number of latent functions (columns of \mathbf{Z}) are unbounded, that is, the IBP prior considers an infinity number of latent functions. The main idea is to infer the number of latent functions when the probabilistic model is conditioned on observed data. To do so, we contribute with a variational approach for Bayesian inference based on the works described in Doshi-Velez et al. (2009) and the variational LFMs (Álvarez et al., 2009). From this approach, we are also able to estimate the interconnection between the latent functions and the outputs. This approach is described in Chapter 3.

1.2.2 Modelling multiple-input multiple-output data

In the standard LFM, we are able to model multiple output data by assuming that the dynamical systems are known before hand. Specifically, the dynamical systems are fully characterized by the impulse response functions (IRF) (e.g. $G(t)$ in figure 1.1). However, in some practical problems we have no knowledge about which differential equations regulated the observed data. Thus, we propose to approximate the unknown IRF by using the orthonormal set of Laguerre functions. Furthermore, Laguerre functions can be encoded within the covariance function of LFMs, and we are able to point-estimate the IRFs using the standard learning process of GPs (i.e. maximizing the logarithm of the marginalized likelihood), or the Kalman filter procedure in sequential LFMs (Hartikainen and Särkkä, 2011). In consequence, we contribute with methods aimed to model multiple-output and multiple-input data with the addition that we are able to point-estimate the IRFs of linear dynamical systems based on the LFM approaches. These methods are described in Chapter 4.

1.2.3 Wiener systems and latent force models

A Wiener system is a non-linear dynamic system, that is build by transforming the response function of a linear dynamic system using a static non-linear function, as shown in Figure 1.4. In Chapter 5, we approximate Wiener systems using standard and se-

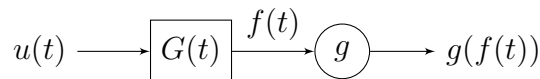


Figure 1.4: Wiener system representation.

quential LFMs. In the first proposed method, we linearise the non-linear static function over the posterior mean of the response function, as described in Steinberg and Bonilla (2014). This model is aimed to infer the posterior of the latent forces as in the standard LFMs.

Additionally in Chapter 5, we propose to approximate the impulse response function of Wiener systems using Laguerre functions and sequential LFMs. To do so, we adopt the Extended Kalman filter at the inference stage (Särkkä, 2013), that allow us to have a tractable model where the non-linear static function is linearised using Taylor series.

1.3 Software and publications

The probabilistic model proposed to estimate the number of latent functions is presented in Chapter 3. This work is based on Guarnizo et al. (2015) and a paper by Guarnizo and Álvarez, which is in review process at the time of writing this thesis. The software developed to reproduce the experiments described in Chapter 3.6 is publicly available at <https://github.com/cdguarnizo/IBPLFM>.

Chapter 4 presents two different models aimed to estimate the impulse response functions of linear dynamic systems. The model focused on the convolution process is based on Guarnizo and Álvarez (2017), and the software aimed to reproduce the experiments is available at https://github.com/cdguarnizo/lag_cgp. On the other hand, the model developed using the state-space model is based on ideas discussed with M. A. Álvarez and S. Sarkka, and the experiments shown in section 4.5 can be reproduced by the software available at <https://github.com/cdguarnizo/lag-irf-slfm>.

Finally, the Wiener system approximations proposed in Chapter 5 are focused on the inference of the latent forces and the impulse response functions. The former approach is based on ideas discussed with M. A. Álvarez, and its code is available at <https://github.com/cdguarnizo/linearizedLFM>. Meanwhile, the latter model is based on ideas discussed with M. A. Álvarez and S. Sarkka, and its code can be found at <https://github.com/cdguarnizo/lag-irf-slfm>.

Note that the codes based on the latent force model use the GPmat toolbox, available at <https://github.com/SheffieldML/GPmat>. On the other hand, the codes based on the sequential latent force model use the LFM toolbox, which is available at <http://becs.aalto.fi/en/research/bayes/lfm/>.

Chapter 2

Linear latent force models

We start by describing linear time invariant (LTI) systems and GPs in section 2.1. Next, we show how to incorporate knowledge from Linear Operators within the covariance function of Gaussian process in section 2.1.3. From the above, we introduce the concept of Latent force models (LFMs) in section 2.2. LFMs can be seen as GPs where the covariance function incorporates the knowledge of a LTI system. This knowledge is induced by the convolution integral which represents the inverse linear operation of differential equations. Practically, we are able to sample random responses of LTI systems by sampling a LFM. Thus, linear LFMs are better suited to explain or model the uncertainty of noisy observations from dynamical systems than using GPs based on general purpose covariance functions.

The original contribution of this chapter is to condense and compare the mathematical foundations of Latent force models.

2.1 Background

2.1.1 Linear dynamical systems

Continuous dynamical systems are usually described by means of differential equations. We start by reviewing LTI systems. LTI systems can be represented by the following ordinary differential equation (ODE) of order $P \in \mathbb{Z}$,

$$\frac{d^P f(t)}{dt^P} + a_{P-1} \frac{d^{P-1} f(t)}{dt^{P-1}} + \dots + a_1 \frac{df(t)}{dt} + a_0 f(t) = u(t), \quad (2.1)$$

where $t \in \mathbb{R}$ is the input time, $f(t)$ is the output function, $a_i \in \mathbb{R}$ weights the i -th derivative of $f(t)$ with respect to t , and $u(t)$ is the forcing function. Here, we review two approaches to solve ODEs. The first one is known as convolution process, and the second one is state-space models. For the following approaches, we assume that observation data is corrupted by an additive noise, as follows

$$y(t) = f(t) + w(t), \quad (2.2)$$

where $w(t)$ follows a zero mean white noise process with variance σ_f^2 .

Convolution process

Notice that in (2.1) the forcing function, $u(t)$, is expressed in terms of the derivatives of $f(t)$. Interestingly, we are able to interchange the roles of these variables by using the following convolution,

$$f(t) = \int_{\mathcal{T}} G(t - \tau) u(\tau) d\tau, \quad (2.3)$$

where \mathcal{T} is the input domain (for the ODE case $\mathcal{T} = \{-\infty, \infty\}$) and $G(t)$ is the impulse response or the Green's function associated to the differential equation (Duffy, 2015). Note that $G(t)$ is calculated by finding the solution of (2.1) when the excitation becomes a delta function (also known as the impulse function). Thus, $G(t)$ depends on parameters a_i and the ODE's order. To find the solution of (2.3), we require to know both mathematical descriptions of $G(t)$ and $u(t)$ (e.g. trigonometric function expressions). However, $u(t)$ may have any form in real-world applications. A block diagram representing the LTI system is shown in figure 2.1, where the rectangular block represents the convolution process described in (2.3).

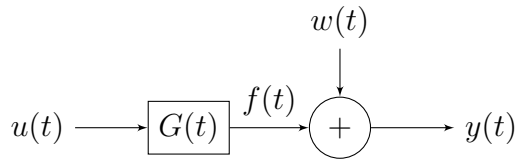


Figure 2.1: Block diagram of a LTI system.

LTI systems can also be modelled by state-space models, as described next.

State-space models

The LTI system given in equation (2.1) can be transformed into a state-space model (Wang, 2009), by defining the state vector as $\mathbf{x}(t) = [f(t), \mathrm{d}f(t)/\mathrm{d}t, \dots, \mathrm{d}^{P-1}f(t)/\mathrm{d}t^{P-1}]^\top$. Then, the state-space model is represented as follows,

$$\frac{\mathrm{d}\mathbf{x}(t)}{\mathrm{d}t} = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t), \quad (2.4)$$

$$f(t) = \mathbf{C}\mathbf{x}(t), \quad (2.5)$$

with

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 1 \\ -a_0 & -a_1 & \dots & \dots & -a_{P-1} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}^\top. \quad (2.6)$$

The form of the matrices $\mathbf{A} \in \mathbb{R}^{P \times P}$, $\mathbf{B} \in \mathbb{R}^P$ and $\mathbf{C} \in \mathbb{R}^P$ is known as the companion form (Wang, 2009).

2.1.2 Gaussian Processes

Gaussian Processes (GPs) are non-parametric probabilistic models that extend multivariate Gaussian distributions to a function space of infinite dimension (Rasmussen and Williams, 2006). To keep the notation consistent across sections, we use t to denote the index (note that the index in general can be multidimensional), and $f(t)$ is a random variable indexed by t . A GP is fully defined by a mean function

$$m(t) = \mathbb{E}[f(t)],$$

and a covariance function

$$k(t, t') = \mathbb{E}[(f(t) - m(t))(f(t') - m(t'))].$$

Thus, the GP can be declared as

$$f(t) \sim \mathcal{GP}(m(t), k(t, t')). \quad (2.7)$$

One of the main features of GPs is when we consider a finite set of index inputs, $\mathbf{t} = [t_1, \dots, t_N]^\top$, then the vector of function values $\mathbf{f} = [f(t_1), \dots, f(t_N)]^\top$ is drawn for a multivariate Gaussian distribution as shown in the following equation

$$\mathbf{f} \sim \mathcal{N}(\mathbf{m}, \mathbf{K}),$$

where \mathbf{m} is a vector composed by evaluations of $m(t)$ at \mathbf{t} , and \mathbf{K} is a squared matrix with elements calculated using $k(t, t')$ at \mathbf{t} . In general, the mean and covariance functions are controlled by hyper parameters. For instance, let us consider the Squared Exponential (SE) and Matérn covariance functions,

$$k_{\text{SE}}(t, t') = s^2 \exp\left(-\frac{(t - t')^2}{2l^2}\right), \quad (2.8)$$

and

$$k_{\text{Matérn}}(t, t') = s^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}(t - t')}{l}\right)^\nu B_\nu\left(\frac{\sqrt{2\nu}(t - t')}{l}\right), \quad (2.9)$$

where $\Gamma(\cdot)$ is the gamma function and $B_\nu(\cdot)$ is a modified Bessel-function of order ν (Rasmussen and Williams, 2006). Note that the overall correlation scale and variability are controlled by the hyperparameters l and s^2 , respectively (Hartikainen and Särkkä, 2010). Interestingly, the hyperparameters play a key in

Now, let us consider a standard regression task where we are given a set of training data points $\{(t_n, y_n)\}_{n=1}^N$. We adopt the relationship defined in equation (2.2), but considering that $f(t)$ follows a GP prior with zero mean and covariance function $k(t, t')$. Thus, we are able to learn the model hyperparameters $\boldsymbol{\theta}_{\text{GP}} = [\sigma_f^2, \boldsymbol{\theta}_{\text{kern}}]$ (where $\boldsymbol{\theta}_{\text{kern}}$ comprises the hyperparameters of the covariance function), by maximizing the logarithm of the marginalised likelihood function, as follows

$$\begin{aligned} \underset{\boldsymbol{\theta}_{\text{GP}}}{\text{maximize}} \quad \log p(\mathbf{y}|\mathbf{t}) = & -\frac{1}{2}\mathbf{y}^\top (\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma_f^2 \mathbf{I}_N)^{-1} \mathbf{y} - \frac{1}{2} \log |\mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma_f^2 \mathbf{I}_N| - \\ & \frac{N}{2} \log(2\pi), \end{aligned} \quad (2.10)$$

with $\mathbf{y} = [y(t_1), \dots, y(t_N)]^\top$. Note that the first term of (2.10) represents the data fitting, while the second term indicates the complexity of the model. This allow us to find a regularized solution that balances between the model complexity and its fit to the data. Furthermore, the standard GP inference process require $O(N^3)$ time to evaluate the above objective function, then it can only be applied to a few thousands

observations. After maximizing (2.10), we are able to predict the unknown set of values $\mathbf{f}^* = [f^*(t_1^*), \dots, f^*(t_{N^*}^*)]^\top$ by conditioning on the training data, as follows

$$\mathbf{f}^* | \mathbf{y} \sim \mathcal{N} \left(\mathbf{K}_{\mathbf{f}^*, \mathbf{f}} [\mathbf{K}_{\mathbf{f}, \mathbf{f}} + \sigma_f^2 \mathbf{I}_N]^{-1} \mathbf{y}, \mathbf{K}_{\mathbf{f}^*, \mathbf{f}^*} - \mathbf{K}_{\mathbf{f}^*, \mathbf{f}} [\mathbf{K}_{\mathbf{f}, \mathbf{f}} + \sigma_f^2 \mathbf{I}_N]^{-1} \mathbf{K}_{\mathbf{f}, \mathbf{f}^*}^\top \right), \quad (2.11)$$

where $\mathbf{K}_{\mathbf{f}^*, \mathbf{f}} \in \mathbb{R}^{N^* \times N}$, with elements calculated using $k_{f,f}(t, t')$. The above posterior is also a GP, where its variance shrinks to test points that are near to the training time values, i.e. this posterior gets confident around the known data.

2.1.3 Gaussian Processes and Linear Operators

In this section, we review the procedure to incorporate knowledge on GPs by using linear operators. An operator \mathcal{L} is considered to be linear if, for any pair of functions $f(t)$ and $g(t)$, and a scalar $\alpha \in \mathbb{R}$ satisfies the following conditions,

$$\mathcal{L}[f(t) + g(t)] = \mathcal{L}[f(t)] + \mathcal{L}[g(t)]$$

and

$$\mathcal{L}[\alpha f(t)] = \alpha \mathcal{L}[f(t)].$$

The most common linear operators are weighted sums, integrals, derivatives and differential equations. Let us start by assuming that the function $f(t)$ follows a GP prior, as

$$f(t) \sim \mathcal{GP}(0, k_{f,f}(t, t')).$$

Furthermore, let us assume that $u(t)$ and $f(t)$ are related by the differential equation defined in (2.1), which can be re-written as $u(t) = \mathcal{L}_t[f(t)]$, where \mathcal{L}_t contains the scaled derivative operations w.r.t. t . Then, we are able to calculate the mean of $u(t)$ as

$$\begin{aligned} m_u(t) &= \mathbb{E}[u(t)] \\ &= \mathbb{E}[\mathcal{L}_t[f(t)]] \\ &= \mathcal{L}_t[\mathbb{E}[f(t)]] \\ &= \mathcal{L}_t[0] \\ &= 0. \end{aligned}$$

And its covariance as

$$\begin{aligned}
k_{uu}(t, t') &= \mathbb{E}[(u(t) - m_u(t))(u(t') - m_u(t'))] \\
&= \mathbb{E}[(u(t) - 0)(u(t') - 0)] \\
&= \mathbb{E}[\mathcal{L}_t[f(t)]\mathcal{L}_{t'}[f(t')]] \\
&= \mathcal{L}_t[\mathcal{L}_{t'}[\mathbb{E}[f(t)f(t')]]] \\
&= \mathcal{L}_t[\mathcal{L}_{t'}[k_{f,f}(t, t')]].
\end{aligned}$$

Thus, the forcing function $u(t)$ is modelled by the following GP prior,

$$u(t) \sim \mathcal{GP}(0, \mathcal{L}_t[\mathcal{L}_{t'}[k_{f,f}(t, t')]]),$$

where its covariance function encodes the knowledge from the linear transformation of $f(t)$. Recall that we are able to express $f(t)$ in function of $u(t)$ by using the convolution integral described in (2.3). In consequence, this convolution can be seen as the inverse operation of \mathcal{L}_t . Furthermore, we can re-write (2.1) as

$$f(t) = \mathcal{L}_t^{-1}[u(t)].$$

Now, if we assume that $u(t) \sim \mathcal{GP}(0, k_{u,u}(t, t'))$, it can be demonstrated that

$$f(t) \sim \mathcal{GP}(0, \mathcal{L}_t^{-1}[\mathcal{L}_{t'}^{-1}[k_{u,u}(t, t')]]). \quad (2.12)$$

As demonstrated above, we are either able to place the basic GP prior over $f(t)$ or $u(t)$, and then, build a new GP prior regarding the linear operation that relates both variables. Hence, we can divide the above process into two main approaches: collocation and latent force methods. In the former, the basic GP prior is placed over the solution/response function $f(t)$, as in Graepel (2003); Raissi et al. (2017). While in the latter, the basic GP prior is instead placed over the forcing function $u(t)$, as in Álvarez et al. (2013); Boyle and Frean (2005); Lawrence et al. (2006).

2.2 Latent force models

LFMs focus on incorporating knowledge about a linear dynamic system within the covariance function of a Gaussian Process. Specifically, LFMs construct a GP prior for the response function $f(t)$ from the GP prior placed over the excitation function $u(t)$

as demonstrated in (2.12). Next, we present the mathematical foundation behind the LFM approach.

2.2.1 Model definition

In a multi-variate regression setting the likelihood model for each output can be expressed as

$$y_d(t) = f_d(t) + w_{f_d}(t), \quad (2.13)$$

where t is the input time, $\{y_d(t)\}_{d=1}^D$ is the collection of D outputs, $w_{f_d}(t)$ is an independent noise process with variance $\sigma_{f_d}^2$, $f_d(t) = \sum_{q=1}^Q f_{d,q}(t)$ and each $f_{d,q}(t)$ is given by

$$f_{d,q}(t) = S_{d,q} \int_{\mathcal{T}} G_d(t - \tau) u_q(\tau) d\tau, \quad (2.14)$$

where $G_d(t)$ is the Green's function associated to the d -th dynamical system, \mathcal{T} is the input time domain, $\{u_q(t)\}_{q=1}^Q$ are latent functions also known as latent forces, and the sensitivities $\mathbf{S} = [S_{d,q}] \in \mathbb{R}^{D \times Q}$ measure the influence of the latent function q over the output d . Additionally, we assume that each latent force $u_q(t)$ is an independent Gaussian process with zero mean function and covariance function $k_{u_q, u_q}(t, t')$.

Note that although the latent forces $\{u_q(t)\}_{q=1}^Q$ are independent, the response functions $\{f_{d,q}(t)\}_{d=1}^D$ are correlated because they depend on the q -th latent force. Thus, we are capable to use this dependency to improve the prediction at any output, by using the information from the other outputs. Next we describe the mathematical foundation of the covariance functions involved in the LFMs framework.

Due to the linearity and the dependency induced by the latent forces in (2.14), the set of processes $\{f_{d,q}(t)\}_{d=1}^D$ follows a joint Gaussian process with mean function equal to zero, and the covariance function between any two response functions is defined as

$$k_{f_d, f_{d'}}^{(q)}(t, t') = \int_{\mathcal{T}} \int_{\mathcal{T}'} G_d(t - \tau) G_{d'}(t' - \tau') k_{u_q, u_q}(\tau, \tau') d\tau d\tau'. \quad (2.15)$$

From the above equation, we are able to find the covariance function between two noiseless outputs as

$$k_{f_d, f_{d'}}(t, t') = \sum_{q=1}^Q S_{d,q} S_{d',q} k_{f_d, f_{d'}}^{(q)}(t, t').$$

Besides the covariance function defined above, we are interested in the cross covariance

function between $f_d(t)$ and $u_q(t)$, which follows

$$k_{f_d, u_q}(t, t') = S_{d,q} \int_{\mathcal{T}} G_d(t - \tau) k_{u_q, u_q}(\tau, t') d\tau. \quad (2.16)$$

For some forms of $G_d(t)$ and the covariance function $k_q(t, t')$, the covariance functions $k_{f_d, f_{d'}}^{(q)}(t, t')$ and $k_{f_d, u_q}(t, t')$ can be found analytically. From now on, we refer to the set of hyperparameters of any covariance function as $\boldsymbol{\theta}_{\text{ker}}$.

Next, we briefly describe three different Green's functions. The first two are obtained from ODEs, meanwhile the third one can be obtained from the Heat equation (Duffy, 2015). For the covariance functions based on ODEs, we assume that $k_{u_q, u_q}(t, t')$ follows a square exponential covariance function, given by

$$k_{u_q, u_q}(t, t') = \exp\left(-\frac{(t - t')^2}{l_q^2}\right), \quad (2.17)$$

where $l_q \in \mathbb{R}^+$ is the length-scale associated to the q -th latent force. Furthermore, the convolution described in (2.14) becomes

$$f_{d,q}(t) = S_{d,q} \int_0^t G_d(t - \tau) u_q(\tau) d\tau, \quad (2.18)$$

given that $G_d(t)$ is defined for $t > 0$, and the initial conditions are assumed to be zero. Additionally, we define the following auxiliary functions

$$h_q(\alpha, \beta, x, z) = \frac{1}{\alpha + \beta} [\Upsilon_q(\beta, x, z) - \exp(-\alpha x) \Upsilon_q(\beta, 0, z)],$$

$$\Upsilon_q(\beta, x, z) = \exp\left(\frac{l_q^2 \beta^2}{4} + \beta(x - z)\right) \left[\operatorname{erf}\left(\frac{x}{l_q} + \frac{l_q \beta}{2}\right) - \operatorname{erf}\left(\frac{x - z}{l_q} + \frac{l_q \beta}{2}\right) \right],$$

where $\operatorname{erf}(\cdot)$ is the error function defined as

$$\operatorname{erf}(t) = \frac{1}{\sqrt{\pi}} \int_0^t \exp(-\tau^2) d\tau.$$

Functions $h_q(\alpha, \beta, x, z)$ and $\Upsilon_q(\beta, x, z)$ appear on the evaluation of the covariance functions based on ODEs, as shown next.

2.2.2 First order ordinary differential equation (ODE1)

In this scenario, we assume that the output can be explained using the following first order ODE,

$$\frac{df_d(t)}{dt} + B_d f_d(t) = \sum_{q=1}^Q S_{d,q} u_q(t), \quad (2.19)$$

where B_d is the decay constant for output d (Lawrence et al., 2006). Now, the solution for the Green's function associated to (2.19) is given by

$$G_d(t) = \exp(-B_d t), \quad (2.20)$$

for $t \geq 0$ and zero otherwise. It can be demonstrated that after replacing (2.20) in (2.15) and (2.16), their closed form are

$$k_{f_d, f_{d'}}^{(q)}(t, t') = \frac{\sqrt{\pi} l_q}{2} [h_q(B_d, B_{d'}, t, t') + h_q(B_{d'}, B_d, t', t)],$$

and

$$k_{f_d, u_q}(t, t') = S_{d,q} \frac{\sqrt{\pi} l_q}{2} \Upsilon_q(B_d, t, t'),$$

respectively.

2.2.3 Second order ordinary differential equation (ODE2)

Here, we assume that the dynamic behaviour of each output is described by a second order differential equation related to a mechanical system as

$$\frac{d^2 f_d(t)}{dt^2} + C_d \frac{df_d(t)}{dt} + B_d f_d(t) = \sum_{q=1}^Q S_{d,q} u_q(t), \quad (2.21)$$

where C_d and B_d are the damper and spring constants for output d , respectively. Without loss of generality, we have assumed that the mass value is one and initial conditions equal to zero, then the solution for the Green's function associated to (2.21) is given by

$$G_d(t) = \frac{1}{\omega_d} \exp\left(-\frac{C_d}{2} t\right) \sin(\omega_d t), \quad (2.22)$$

for $t \geq 0$ and zero otherwise. Where ω_d is the natural frequency of (2.21) and it is defined as $\omega_d = \sqrt{4B_d - C_d^2/2}$ (Álvarez et al., 2013). Besides, it can be demonstrated that after replacing (2.22) in (2.15), the covariance function between two response functions

becomes

$$k_{f_d, f_{d'}}^{(q)}(t, t') = K_0 [h_q(\gamma_{d'}, \tilde{\gamma}_d, t, t') - h_q(\gamma_{d'}, \gamma_d, t, t') + h_q(\tilde{\gamma}_{d'}, \gamma_d, t, t') - h_q(\tilde{\gamma}_{d'}, \tilde{\gamma}_d, t, t') \\ + h_q(\gamma_d, \tilde{\gamma}_{d'}, t', t) - h_q(\gamma_d, \gamma_{d'}, t', t) + h_q(\tilde{\gamma}_d, \gamma_{d'}, t', t) - h_q(\tilde{\gamma}_d, \tilde{\gamma}_{d'}, t', t)]$$

with

$$K_0 = \frac{\sqrt{\pi} l_q}{8\omega_d \omega_{d'}}, \quad \gamma_d = \alpha_d + j\omega_d, \quad \tilde{\gamma}_d = \alpha_d - j\omega_d \\ \omega_d = \frac{\sqrt{4B_d - C_d^2}}{2}, \quad \alpha_d = \frac{C_d}{2},$$

where ω_d is the natural frequency and $j = \sqrt{-1}$. Furthermore, the cross-covariance, defined in (2.16), reduces to

$$k_{f_d, u_q}(t, z) = \frac{l_q S_{d,q} \sqrt{\pi}}{j4\omega_d} [\Upsilon_q(\tilde{\gamma}_d, t, z) - \Upsilon_q(\gamma_d, t, z)]. \quad (2.23)$$

2.2.4 Gaussian smoothing (GS)

We present a general purpose covariance function that has been successfully used for the explanation of several multi-task regression problems in [Álvarez et al. \(2009\)](#); [Zhao and Sun \(2014\)](#). We start by assuming that both $G_d(t - t')$ and $k_{u_q}(t, t')$ have the following form

$$k_{u_q, u_q}(t, t') = \left(\frac{\beta_{u_q}}{2\pi} \right)^{1/2} \exp \left(-\frac{\beta_{u_q}}{2} [t - t']^2 \right),$$

where β_{u_q} is the precision value and $t \in \mathbb{R}$. This Green's function can be found from the Heat or Wave equations ([Duffy, 2015](#)). Then, it can be shown that the covariance function $k_{f_d, f_{d'}}^{(q)}(t, t')$ is defined as

$$k_{f_d, f_{d'}}^{(q)}(t, t') = \frac{1}{(2\pi p_{d,d'}^{(q)})^{1/2}} \exp \left(-\frac{1}{2p_{d,d'}^{(q)}} [t - t']^2 \right),$$

with $p_{d,d'}^{(q)} = p_{f_d}^{-1} + p_{f_{d'}}^{-1} + p_{u_q}^{-1}$. Parameters p_{f_d} and p_{u_q} correspond to the precision values associated to $G_d(t)$ and $k_{u_q, u_q}(t, t')$, respectively.

2.2.5 Inference

Let us assume that we are given a dataset consisting of the corrupted response observations of D outputs, declared as $\{\mathbf{y}_d\}_{d=1}^D$ with $\mathbf{y}_d \in \mathbb{R}^{N_d}$. Additionally, each \mathbf{y}_d is indexed by time vectors $\{\mathbf{t}_d\}_{d=1}^D$ with $\mathbf{t}_d \in \mathbb{R}^{N_d}$, and N_d is the number of data points associated to output d . We adopt the relationship defined in equation (2.13), where $f_d(t)$ follows a LFM prior with zero mean and covariance function $k_{f,f}(t, t')$. We start by defining the model hyperparameters $\boldsymbol{\theta}_{\text{LFM}} = [\boldsymbol{\sigma}_f, \boldsymbol{\theta}_{\text{kern}}]$, where $\boldsymbol{\theta}_{\text{kern}}$ comprises the hyperparameters of all the covariance functions, and $\boldsymbol{\sigma}_f$ contains the D noise variance values. Then, we learn the hyperparameters by maximizing the logarithm of the marginalised likelihood function, as follows

$$\begin{aligned} \underset{\boldsymbol{\theta}_{\text{LFM}}}{\text{maximize}} \quad \log p(\mathbf{y}|\mathbf{t}) = & -\frac{1}{2}\mathbf{y}^\top (\mathbf{K}_{f,f} + \boldsymbol{\Sigma}_f)^{-1} \mathbf{y} - \frac{1}{2} \log |\mathbf{K}_{f,f} + \boldsymbol{\Sigma}_f| \\ & - \frac{N}{2} \log(2\pi), \end{aligned} \quad (2.24)$$

with $\mathbf{y} = [\mathbf{y}_1^\top, \dots, \mathbf{y}_D^\top]^\top$, $\mathbf{t} = [\mathbf{t}_1^\top, \dots, \mathbf{t}_D^\top]^\top$, $N = \sum_{d=1}^D N_d$, and $\mathbf{K}_{f,f}$ is a block-wise matrix, where the elements of the block located at the d -th row and d' -th column are evaluated using $k_{f_d, f_{d'}}(t, t')$. Similarly, $\boldsymbol{\Sigma}_f$ is diagonal block matrix, where its d -th block is given by $\sigma_{f_d}^2 \mathbf{I}_{N_d}$.

2.2.6 Predictive distributions

As shown in (2.11), we are able to predict values of $y_d(t)$ and $f_d(t)$ at unknown time values $\mathbf{t}^* = [\mathbf{t}_1^{*\top}, \dots, \mathbf{t}_D^{*\top}]^\top$ using

$$\mathbf{y}^*|\mathbf{y} \sim \mathcal{N}\left(\mathbf{K}_{f^*,f}[\mathbf{K}_{f,f} + \boldsymbol{\Sigma}_f]^{-1}\mathbf{y}, \mathbf{K}_{f^*,f^*} - \mathbf{K}_{f^*,f}[\mathbf{K}_{f,f} + \boldsymbol{\Sigma}_f]^{-1}\mathbf{K}_{f^*,f}^\top + \boldsymbol{\Sigma}_{f^*}\right) \quad (2.25)$$

and

$$\mathbf{f}^*|\mathbf{y} \sim \mathcal{N}\left(\mathbf{K}_{f^*,f}[\mathbf{K}_{f,f} + \boldsymbol{\Sigma}_f]^{-1}\mathbf{y}, \mathbf{K}_{f^*,f^*} - \mathbf{K}_{f^*,f}[\mathbf{K}_{f,f} + \boldsymbol{\Sigma}_f]^{-1}\mathbf{K}_{f^*,f}^\top\right), \quad (2.26)$$

respectively. $\mathbf{K}_{f^*,f}$ is a $D \times D$ block-wise matrix, where the elements of the block located at the d -th row and d' -th column are calculated with $k_{f_d, f_{d'}}(t^*, t')$. $\boldsymbol{\Sigma}_{f^*}$ is a block diagonal matrix, where the elements of the d -th block are calculated using $\sigma_{f_d}^2 \mathbf{I}_{N_d^*}$, and N_d^* is the number of test points at output d .

Interestingly, we are able to calculate the posterior of the latent forces $\mathbf{u}^* = [\mathbf{u}_1^{*\top}, \dots,$

$\mathbf{u}_Q^{*\top}]^\top$, similarly to (2.26), as

$$\mathbf{u}^*|\mathbf{y} \sim \mathcal{N}\left(\mathbf{K}_{\mathbf{u}^*,\mathbf{f}}[\mathbf{K}_{\mathbf{f},\mathbf{f}} + \boldsymbol{\Sigma}_{\mathbf{f}}]^{-1}\mathbf{y}, \mathbf{K}_{\mathbf{u}^*,\mathbf{u}^*} - \mathbf{K}_{\mathbf{u}^*,\mathbf{f}}[\mathbf{K}_{\mathbf{f},\mathbf{f}} + \boldsymbol{\Sigma}_{\mathbf{f}}]^{-1}\mathbf{K}_{\mathbf{f},\mathbf{u}^*}\right). \quad (2.27)$$

Hence, we are able find the predictive distribution for the inverse problem using the LFM formulation. The inverse problem is related to finding the function $u(t)$ and ODE's parameters solely from the output data. Note that the ODE's parameters are point-estimated by the maximization of the marginal likelihood.

2.3 Sequential Latent Force models

LFMs can be represented as a linear state-space model driven by a white noise process (Hartikainen and Särkkä, 2011). Furthermore, this state-space model can be extended to consider different non-linear scenarios as: Wiener, Hammerstein and Drift models (Hartikainen et al., 2012). We start by defining the sequential model used to approximate a GP prior.

2.3.1 Sequential Gaussian Process

Sequential Gaussian Process (SGP) aims to represent the random function $u(t)$ with covariance function $k_{u,u}(t, t')$ by means of a LTI stochastic differential equation (Hartikainen and Särkkä, 2010). Thus, we are able to model $u(t)$ as

$$\begin{aligned} \frac{d\mathbf{x}_u(t)}{dt} &= \mathbf{A}_u\mathbf{x}_u(t) + \mathbf{B}_u\epsilon(t), \\ u(t) &= \mathbf{C}_u\mathbf{x}_u(t), \end{aligned} \quad (2.28)$$

where $\mathbf{x}_u(t) = [u(t), du(t)/dt, \dots, d^{K-1}u(t)/dt^{K-1}]^\top$ comprises $u(t)$ and its derivatives w.r.t. t , and $\epsilon(t) \sim \mathcal{N}(0, \sigma_\epsilon^2)$. Matrices \mathbf{A}_u , \mathbf{B}_u and \mathbf{C}_u follow the same form as defined in (2.6). The above model can be seen as a filtering process over the noise process $\epsilon(t)$. In consequence, $u(t)$ represents a smoothed version of $\epsilon(t)$. Furthermore, the above model is able to straightforwardly represent a GP prior, if the spectral density of the covariance function has the following form

$$S(\omega) = \frac{\text{constant}}{\text{polynomial in } \omega^2},$$

where ω is the frequency variable. Spectral density of Matérn class covariance functions fulfils the above requirement, but for the Squared Exponential covariance function a Taylor series approximation should be adopted, as shown in [Hartikainen and Särkkä \(2010\)](#).

2.3.2 Model definition

The GP prior defined in (2.28) can be augmented in order to consider multiple responses of LTI systems from multiple excitations ([Hartikainen and Särkkä, 2011](#)), as follows

$$\begin{aligned}\frac{d\mathbf{x}(t)}{dt} &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\boldsymbol{\epsilon}(t), \\ \mathbf{f}(t) &= \mathbf{C}\mathbf{x}(t),\end{aligned}\tag{2.29}$$

with $\mathbf{f}(t) = [f_1(t), \dots, f_D(t)]^\top$, $\boldsymbol{\epsilon}(t) = [\epsilon_1(t), \dots, \epsilon_Q(t)]^\top$, $\boldsymbol{\epsilon}(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_\epsilon)$, $\mathbf{x}(t) = [\mathbf{x}_f(t)^\top, \mathbf{x}_u(t)^\top]^\top$, $\mathbf{x}_f(t)$ and $\mathbf{x}_u(t)$ comprises the derivatives w.r.t. t of $\{f_d(t)\}_{d=1}^D$ and $\{u_q(t)\}_{q=1}^Q$, respectively. Matrices \mathbf{A} , \mathbf{B} and \mathbf{C} are constructed such that they operate appropriately on the augmented state-space model. For example, the second order LFM described in (2.21), with Q and D equal to one, can be represented using the above state-space model by setting

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -B_1 & -C_1 & -S_{1,1} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -a_0 & -a_1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix},$$

and $\mathbf{C} = [1, 0, 0, 0]$. Let us recall that coefficients a_i 's are used for the approximation of the GP prior over $u(t)$. From now on, we refer to the model defined in (2.29) as Sequential Latent Force Model (SLFM). Next, we briefly review the procedures used to estimate the SLFM's parameters.

2.3.3 Sequential inference

In order to learn the parameters of SLFMs, we convert the continuous model, given in (2.29), into the following discrete model,

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{q}_{k-1}, \quad \mathbf{q}_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}),$$

where the transition process and the covariance matrix related to the noise process can be solved on sampling time T as

$$\mathbf{F} = \exp(\mathbf{A}T), \quad \mathbf{Q} = \int_0^T \exp(\mathbf{A}\tau) \mathbf{Q}_\epsilon \exp(\mathbf{A}\tau) d\tau.$$

Note that $\mathbf{Q}_\epsilon = \text{diag}([\sigma_{\epsilon_1}, \dots, \sigma_{\epsilon_Q}])$ is the covariance matrix of $\epsilon(t)$, and the above exponential evaluations are calculated using the exponential matrix. Additionally, we assume that the observation data is modelled as

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{r}_k, \quad \mathbf{r}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k),$$

where \mathbf{x}_k represents the evaluation $\mathbf{x}(t_k)$, \mathbf{r}_k represents the noise process at the observations, and t_k is multiple of the sampling period T , as kT . Typically, to avoid an exponential increment of the full posterior of the states regarding the number of observations, the dynamic model is restricted to follow a probabilistic Markov sequence (Särkkä, 2013). Hence, the linear discrete state-space model is described by

$$\mathbf{x}_0 \sim p(\mathbf{x}_0), \quad \mathbf{x}_k \sim p(\mathbf{x}_k | \mathbf{x}_{k-1}), \quad \mathbf{y}_k \sim p(\mathbf{y}_k | \mathbf{x}_k), \quad (2.30)$$

where \mathbf{x}_0 is the initial (hidden) state-space vector, $p(\mathbf{x}_k | \mathbf{x}_{k-1})$ describes the system dynamics, and $p(\mathbf{y}_k | \mathbf{x}_k)$ represents the measurement model. If we assume that the dynamical system is described by a linear Gaussian state-space model, then, the posterior distribution for the state vector is Gaussian distributed and a closed form solution can be found using the Kalman filter (KF) or the Rauch-Tung-Striebel (RTS) smoother (Särkkä, 2013). Thus, the state predictive and filtering distributions, and the predictive distribution at the k -th step are given by

$$\begin{aligned} p(\mathbf{x}_k | \mathbf{y}_{1:k-1}) &= \mathcal{N}(\mathbf{x}_k | \mathbf{m}_k^-, \mathbf{P}_k^-), \\ p(\mathbf{x}_k | \mathbf{y}_{1:k}) &= \mathcal{N}(\mathbf{x}_k | \mathbf{m}_k, \mathbf{P}_k), \\ p(\mathbf{y}_k | \mathbf{y}_{1:k-1}) &= \mathcal{N}(\mathbf{y}_k | \mathbf{C}\mathbf{m}_k^-, \mathbf{S}_k), \end{aligned}$$

respectively. The means and covariances \mathbf{m}_k^- , \mathbf{m}_k , \mathbf{P}_k^- and \mathbf{P}_k are calculated by the Kalman filter recursive algorithm given in Algorithm 1. Besides, the total computational time of the standard KF is $O(N(P + K)^2)$, where K and P are the order of the states space vectors used to model $u(t)$ and $f(t)$, respectively.

On the other hand, the smoothing distributions can be obtained using the RTS

Algorithm 1 Kalman filter procedure.

```

1: Input:  $\mathbf{y}_{1:N}$ 
2: Initial state  $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{m}_0, \mathbf{P}_0)$ .
3: for  $k \in \{1, \dots, N\}$  do
4:   Prediction:
5:    $\mathbf{m}_k^- = \mathbf{A}_{k-1} \mathbf{m}_{k-1}$  {State predictive mean}
6:    $\mathbf{P}_k^- = \mathbf{A}_{k-1} \mathbf{P}_{k-1} \mathbf{A}_{k-1}^\top + \mathbf{Q}_{k-1}$  {State predictive covariance}
7:   Update:
8:    $\mathbf{e}_k = \mathbf{y}_k - \mathbf{C} \mathbf{m}_k^-$  {Observation error}
9:    $\mathbf{S}_k = \mathbf{C} \mathbf{P}_k^- \mathbf{C}^\top + \mathbf{R}_k$  {Observation variance}
10:   $\mathbf{K}_k = \mathbf{P}_k^- \mathbf{C}^\top \mathbf{S}_k^{-1}$  {Kalman gain}
11:   $\mathbf{m}_k = \mathbf{m}_k^- + \mathbf{K}_k \mathbf{e}_k$  {Filter mean}
12:   $\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^\top$  {Filter variance}
13: end for

```

smoother, which is based on the filtering distributions found using algorithm 1. The smoothing marginal distribution for the k -th state given all the measurements is defined as

$$p(\mathbf{x}_k | \mathbf{y}_{1:N}) = \mathcal{N}(\mathbf{x}_k | \mathbf{m}_k^s, \mathbf{P}_k^s),$$

with moments given by the backwards procedure described in algorithm 2 (Särkkä, 2013). Hyperparameters can be learned by maximizing the logarithm of the marginal

Algorithm 2 RTS smoothing procedure.

```

1: Input:  $\{\mathbf{m}_k, \mathbf{P}_k\}_{k=1}^N$  calculated using algorithm 1.
2: for  $k \in \{N-1, \dots, 1\}$  do
3:    $\mathbf{m}_{k+1}^- = \mathbf{A}_k \mathbf{m}_k$  {State predictive mean}
4:    $\mathbf{P}_{k+1}^- = \mathbf{A}_k \mathbf{P}_k \mathbf{A}_k^\top + \mathbf{Q}_k$  {State predictive covariance}
5:    $\mathbf{G}_k = \mathbf{P}_k \mathbf{A}_k [\mathbf{P}_{k+1}^-]^{-1}$  {Smoother gain}
6:    $\mathbf{m}_k^s = \mathbf{m}_k + \mathbf{G}_k [\mathbf{m}_{k+1}^s - \mathbf{m}_{k+1}^-]$  {Smoother mean}
7:    $\mathbf{P}_k^s = \mathbf{P}_k - \mathbf{G}_k [\mathbf{P}_{k+1}^s - \mathbf{P}_{k+1}^-] \mathbf{G}_k^\top$  {Smoother variance}
8: end for

```

likelihood (Särkkä, 2013), with the objective defined as

$$\underset{\boldsymbol{\theta}_{\text{SLFM}}}{\text{maximize}} \quad \log p(\mathbf{y}_{1:N}) = -\frac{1}{2} \sum_{k=1}^N \left[\log |2\pi \mathbf{S}_k| + \mathbf{e}_k^\top \mathbf{S}_k^{-1} \mathbf{e}_k \right], \quad (2.31)$$

where $\boldsymbol{\theta}_{\text{SLFM}}$ comprises the hyperparameters required to describe the model defined in

(2.29). Note that the k -th sum term from the objective function defined in (2.31) is calculated at the k -th evaluation of the Kalman filter procedure.

2.3.4 Predictive distributions

We are able to predict the value $\mathbf{y}(t^*)$ by including the test time t^* in the Kalman filter and smoothing steps. Hence, the moments of the prediction are given by

$$\mathbb{E}[\mathbf{y}(t^*)] = \mathbf{C}\mathbf{m}_{t^*}, \text{ and } V[\mathbf{y}(t^*)] = \mathbf{S}_{t^*},$$

where \mathbf{m}_{t^*} is the mean vector of the state vector, and \mathbf{S}_{t^*} is the observation covariance matrix. Both matrices are evaluated at time t^* .

2.4 Comparison of Latent Force model approaches

2.4.1 Discrete time or Continuous time

Note that LFMs are fully continuous processes and the correlation among the whole data (covariance matrix) is used to predict test values at any time. Hence, LFMs are learned using the information given by all training points simultaneously.

Meanwhile in the SLFM approach, we are only able to train the model and predict test values at specific time stamps given by the sampling period T . Furthermore, the prediction of \mathbf{x}_k , in the best case scenario, uses the information from \mathbf{x}_{k-1} (KF) and \mathbf{x}_{k+1} (RTS), if available (update step).

2.4.2 Computational complexity

For a dataset consisting of N data points, $O(N^3)$ time is required for the LFM approach during each step of its training phase. However, this computational time can be reduced to $O(NM^2)$ using *inducing variables* (Álvarez et al., 2009), where M refers to the number of inducing variables used to approximate the posterior of the latent forces. Additionally, LFMs can be scaled to large data scenarios as demonstrated in Dai et al. (2014); Gal et al. (2014).

On the other hand, the KF procedure requires $O(N(P + K)^3)$ time, if and only if, the training data points are in sequence and sampled at a constant time rate. Thus, if the dataset is non-uniform sampled or includes missing values, then this cost time will be increased.

In summary, the computational burden of LFMs highly depends on the number of data points. Meanwhile, the cost time in SLFMs is mainly driven by the number of outputs and inputs (length of the state vector), and how the data is sampled.

2.4.3 Non-linear dynamic systems

The first non-linear dynamic system approximated using the LFM framework is introduced in [Lawrence et al. \(2006\)](#). Specifically, gene expression data is explained using a Hammerstein system. Nevertheless, the convolution operation is approximated by sums in order to keep the inference process tractable. In contrast, we are able to build all types of non-linear dynamic systems using SLFMs. As demonstrated in [Hartikainen et al. \(2012\)](#), Wiener, Hammerstein and Drift systems can be approximated using the SLFM approach.

Chapter 3

Automatic selection of the number of latent forces

As mentioned in Chapter 2, LFMs are a powerful tool for modelling data generated by multi-output linear dynamical systems. LFMs are based on the convolution process between the dynamical system's impulse response function (which fully characterizes the linear system) and a set of shared latent functions (where each latent function follows a Gaussian Process prior). Note that we are focused in the standard LFM based on the convolution process, then the Sequential LFM is not considered in this chapter. Within the LFM framework, a fixed quantity of shared latent functions is used to describe multiple-output data. In consequence, the dependency induced by the set of shared latent functions over the outputs can be used to improve prediction tasks. Usually, the number of latent functions is selected depending on the experimenter assumptions or by the prior knowledge available about the data. However, in real-world applications, the number of latent functions is unknown and the interconnection between the outputs and the latent functions might be sparse. Thus, in order to automatically select the number of latent functions, we develop an automated variational method based on the Indian Buffet Process prior which induces sparsity through the network formed by the outputs and the latent functions. The proposed variational method is tested on synthetic, gene expression, movement capture and weather datasets. Results indicate that the proposed model achieves better performance compared with other existing methods.

3.1 Indian Buffet Process

The IBP is a distribution over binary matrices with a finite number of rows and an unbounded number of columns (Griffiths and Ghahramani, 2005). This can define a non-parametric latent feature model in which rows are related to data points and columns are related to latent features. Thus, the relationship between latent features and data points can be encoded in a binary matrix $\mathbf{Z} = [Z_{d,q}] \in \{0, 1\}^{D \times Q}$ with $Q \rightarrow \infty$. Besides, if $Z_{d,q} = 1$ then feature q is used to explain data point d . Each element $Z_{d,q}$ is sampled from the following hierarchical model

$$v_j \sim \text{Beta}(\alpha, 1), \quad \pi_q = \prod_{j=1}^q v_j, \quad Z_{d,q} \sim \text{Bernoulli}(\pi_q), \quad (3.1)$$

where α is a real positive value, and π_q is the probability of observing a non-zero value at the q -th column of the matrix \mathbf{Z} , that is, the value π_q controls the sparsity pattern of the q -th latent feature.

3.2 Model definition

Let us assume we are given a dataset consisting of D outputs, as $\{\mathbf{y}_d\}_{d=1}^D$ with $\mathbf{y}_d \in \mathbb{R}^{N_d}$. Additionally, the outputs are indexed by time vectors $\{\mathbf{t}_d\}_{d=1}^D$ with $\mathbf{t}_d \in \mathbb{R}^{N_d}$, and N_d is the number of data points associated to output d . Regarding the LFM model described in (2.13) and (2.14), and including the IBP prior variable defined in (3.1), we re-define the model given in (2.13), as

$$y_d(t) = \bar{f}_d(t) + w_{f_d}(t), \quad (3.2)$$

with $\bar{f}_d(t) = \sum_{q=1}^{Q \rightarrow \infty} Z_{d,q} f_{d,q}(t)$. Note that the binary variable $Z_{d,q}$ plays a key role by deciding if the q -th latent force contributes in the description of the d -th output. Thus, the above expression leads to the following likelihood function,

$$p(\mathbf{y}|\mathbf{F}, \mathbf{Z}, \mathbf{t}) = \prod_{d=1}^D \mathcal{N} \left(\mathbf{y}_d \left| \sum_{q=1}^{Q \rightarrow \infty} Z_{d,q} \mathbf{f}_{d,q}, \boldsymbol{\Sigma}_{\mathbf{f}_d} \right. \right), \quad (3.3)$$

where $\mathbf{y} = [\mathbf{y}_1^\top, \dots, \mathbf{y}_D^\top]^\top$ and $\mathbf{t} = [\mathbf{t}_1^\top, \dots, \mathbf{t}_D^\top]^\top \in \mathbb{R}^N$ are the stacked versions of the observed data, and $N = \sum_{d=1}^D N_d$ is the total number of data points. Furthermore, $\boldsymbol{\Sigma}_{\mathbf{f}_d} = \beta_d^{-1} \mathbf{I}_{N_d}$ is the noise covariance matrix, β_d is the noise precision value, and \mathbf{I}_{N_d} is

the identity matrix of size N_d . While, $\mathbf{F} = \{\mathbf{f}_{d,q}\}_{d=1,q=1}^{D,\infty}$ is the collection of the dynamic responses from each latent force, with probability defined as

$$p(\mathbf{F}|\mathbf{t}) = \prod_{q=1}^{Q \rightarrow \infty} \mathcal{N}(\mathbf{f}_{:,q} | \mathbf{0}, \mathbf{K}_{\mathbf{f},\mathbf{f}}^{(q)}), \quad (3.4)$$

where $\mathbf{f}_{:,q} = [\mathbf{f}_{1,q}^\top, \dots, \mathbf{f}_{D,q}^\top]^\top \in \mathbb{R}^N$, $\mathbf{K}_{\mathbf{f},\mathbf{f}}^{(q)} \in \mathbb{R}^{N \times N}$ is the covariance matrix of the noiseless outputs given by the q -th latent force and its elements are evaluated using $k_{f_d, f_{d'}}^{(q)}(t, t')$. From (3.1), (3.3) and (3.4) the joint probability of our model is defined as follows

$$p(\mathbf{y}, \mathbf{F}, \mathbf{Z}, \mathbf{v}) = p(\mathbf{y}|\mathbf{F}, \mathbf{Z})p(\mathbf{F})p(\mathbf{Z}|\mathbf{v})p(\mathbf{v}). \quad (3.5)$$

Note that the above probability distributions are evaluated using the hyperparameters $\boldsymbol{\theta} = \{\alpha, \boldsymbol{\beta}, \boldsymbol{\theta}_{\text{kern}}\}$, where $\boldsymbol{\beta}$ is the collection of noise precision values.

3.3 Variational Inference approach

We are interested in learning the probabilistic model defined above and estimating the posterior distributions for the latent forces and \mathbf{Z} . Unfortunately, these posterior distributions are intractable due to the IBP prior. Thus, we resort to approximate them using variational inference methods. Typically, a truncated posterior for \mathbf{Z} is used in variational inference approaches involving the IBP prior, as in [Doshi-Velez et al. \(2009\)](#).

Using an IBP as a prior for a linear Gaussian model, the authors in [Doshi-Velez et al. \(2009\)](#), derived two variational mean field approximations, referred to as “finite variational approach” and “infinite variational approach”. For our variational approximation, we adopt the latter approach, this is, the update equations are based on the true IBP prior over an infinite number of features, but for practical implementation, we use a level of truncation Q_+ as the maximum number of latent functions.

In order to include the inference over the latent forces and reduce the complexity time, we augment the model using inducing variables $\mathbf{u}_q \in \mathbb{R}^M$ (as in [Álvarez et al. \(2009\)](#); [Titsias \(2009\)](#)), which are obtained when evaluating the latent force u_q at a set of M inducing inputs $\boldsymbol{\lambda}_q = [\lambda_{q,1}, \dots, \lambda_{q,M}]^\top$. We refer to the set of inducing variables as $\mathbf{u} = \{\mathbf{u}_q\}_{q=1}^{Q_+}$ and the set of inducing points as $\boldsymbol{\lambda} = \{\boldsymbol{\lambda}_q\}_{q=1}^{Q_+}$. Thus, the prior over \mathbf{F} in

(3.4) is changed to the following conditional distribution

$$p(\mathbf{F}|\mathbf{u}) = \prod_{d=1}^D \prod_{q=1}^{Q_+} \mathcal{N}(\mathbf{f}_{d,q} | \mathbf{K}_{\mathbf{f}_d, \mathbf{u}_q} \mathbf{K}_{\mathbf{u}_q, \mathbf{u}_q}^{-1} \mathbf{u}_q, \mathbf{K}_{\mathbf{f}_d, \mathbf{f}_d}^{(q)} - \mathbf{K}_{\mathbf{f}_d, \mathbf{u}_q} \mathbf{K}_{\mathbf{u}_q, \mathbf{u}_q}^{-1} \mathbf{K}_{\mathbf{u}_q, \mathbf{f}_d}),$$

where $\mathbf{K}_{\mathbf{f}_d, \mathbf{u}_q} \in \mathbb{R}^{N_d \times M}$ is the cross-covariance matrix between $f_d(t)$ and $u_q(t)$, with elements given by $k_{f_d, u_q}(t, t')$. Additionally, the prior over \mathbf{u} has the following form

$$p(\mathbf{u}) = \prod_{q=1}^{Q_+} \mathcal{N}(\mathbf{u}_q | \mathbf{0}, \mathbf{K}_{\mathbf{u}_q, \mathbf{u}_q}),$$

where $\mathbf{K}_{\mathbf{u}_q, \mathbf{u}_q} \in \mathbb{R}^{M \times M}$ is the covariance matrix between $u(\lambda)$ and $u_q(\lambda')$, where the elements are calculated using $k_q(t, t')$. Hence, the joint probability given in (3.5) is augmented as

$$p(\mathbf{y}, \mathbf{F}, \mathbf{u}, \mathbf{Z}, \mathbf{v}) = p(\mathbf{y}|\mathbf{F}, \mathbf{Z})p(\mathbf{F}|\mathbf{u})p(\mathbf{u})p(\mathbf{Z}|\mathbf{v})p(\mathbf{v}).$$

For the proposed variational approach, we adopt the mean field approximation (Bishop, 2006), which assumes that the distribution for the variational variables are independent. Thus, the approximated posterior distribution is denoted as $q(\mathbf{F}, \mathbf{u}, \mathbf{Z}, \mathbf{v}) = q(\mathbf{F}|\mathbf{u})q(\mathbf{u})q(\mathbf{Z})q(\mathbf{v})$, where the variational distributions for \mathbf{Z} and \mathbf{v} factorize as (Doshi-Velez et al., 2009)

$$q(\mathbf{Z}) = \prod_{d=1}^D \prod_{q=1}^{Q_+} q(Z_{d,q}), \quad q(\mathbf{v}) = \prod_{q=1}^{Q_+} q(v_q),$$

respectively. Additionally, we can assume that \mathbf{u} is a sufficient statistic for \mathbf{F} , as in Titsias (2009). Thus, the optimal form for $q(\mathbf{F}|\mathbf{u})$ is given by the true posterior $p(\mathbf{F}|\mathbf{u})$, as $q(\mathbf{F}|\mathbf{u}) = p(\mathbf{F}|\mathbf{u})$. In the mean-field variational inference method, the Kullback-Leibler distance between the variational distribution $q(\mathbf{F}, \mathbf{u}, \mathbf{Z}, \mathbf{v})$ and the true posterior $p(\mathbf{F}, \mathbf{u}, \mathbf{Z}, \mathbf{v}|\mathbf{y})$ is minimized by maximizing the following lower bound (Bishop, 2006)

$$\mathcal{F} = \int q(\mathbf{F}, \mathbf{u}, \mathbf{Z}, \mathbf{v}) \log \left(\frac{p(\mathbf{y}, \mathbf{F}, \mathbf{u}, \mathbf{Z}, \mathbf{v})}{q(\mathbf{F}, \mathbf{u}, \mathbf{Z}, \mathbf{v})} \right) d\mathbf{F} d\mathbf{u} d\mathbf{Z} d\mathbf{v}.$$

After expanding the logarithm of the fraction in the above expression, the lower bound

becomes

$$\begin{aligned}\mathcal{F} = & \mathbb{E} [\log p(\mathbf{y}|\mathbf{F}, \mathbf{Z})p(\mathbf{u})] + \mathbb{E} [\log p(\mathbf{Z}|\mathbf{v})] \\ & + \mathbb{E} [\log p(\mathbf{v})] + H(\mathbf{u}) + H(\mathbf{Z}) + H(\mathbf{v}),\end{aligned}\quad (3.6)$$

where $H(q(x)) = -\int q(x) \ln(q(x))dx$ represents the Shannon entropy. In order to keep the notation uncluttered, all the expected values considered in this Chapter are calculated with respect to the posterior distribution $q(\mathbf{f}, \mathbf{u}, \mathbf{Z}, \mathbf{v})$. Besides, the above expected values are fully described in appendix B.1.

The variational lower bound, given in (3.6), is maximized using an Expectation-Maximization (EM) algorithm (Bishop, 2006), where the E-step consists of updating the moments or parameters of the variational distributions while the hyperparameters are kept fixed. Similarly, the M-step focuses on finding the values of the hyperparameters that maximizes (3.6) while keeping the value of the posteriors' parameters fixed.

Before starting the derivation of the variational distributions, we define the following variables as $\xi_{d,q} = \text{tr}(\mathbf{K}_{\mathbf{f}_d, \mathbf{f}_d}^{(q)})$, $\boldsymbol{\psi}_{d,q} = \mathbf{K}_{\mathbf{f}_d, \mathbf{u}_q}^\top \mathbf{y}_d$, and $\boldsymbol{\Psi}_{d,q,q'} = \mathbf{K}_{\mathbf{f}_d, \mathbf{u}_q}^\top \mathbf{K}_{\mathbf{f}_d, \mathbf{u}_{q'}}$. In fact, their entries are calculated as

$$\begin{aligned}\xi_{d,q} &= \sum_{j=1}^{N_d} k_{f_d, f_d}^{(q)}(t_{d,j}, t_{d,j}), \\ [\boldsymbol{\psi}_{d,q}]_i &= \sum_{j=1}^{N_d} k_{f_d, u_q}(t_{d,j}, \lambda_{q,i}) y_{d,j}, \\ [\boldsymbol{\Psi}_{d,q,q'}]_{i,k} &= \sum_{j=1}^{N_d} k_{f_d, u_q}(t_{d,j}, \lambda_{q,i}) k_{f_d, u_{q'}}(t_{d,j}, \lambda_{q',k}),\end{aligned}$$

where $[\boldsymbol{\psi}]_i$ represents the i -th element of column vector $\boldsymbol{\psi}$, and $[\boldsymbol{\Psi}]_{i,k}$ represents the element at the i -th row and k -th column of matrix $\boldsymbol{\Psi}$. Note that the above variables involve the evaluation of covariance functions over the output data points, and they are calculated as sums along the output data length. Thus, if N_d is considerably large we are capable of caching intermediate sums. In consequence, the proposed EM-algorithm can be scaled in order to deal with large datasets (Dai et al., 2014; Gal et al., 2014).

Continuing with the derivation of the variational approximation, updates for the moments of each variational distribution are given next.

3.3.1 Updates for $q(v_q)$

Note that the following variational approach for v_q is also given in [Doshi-Velez et al. \(2009\)](#). The variational distribution for v_q is assumed as

$$q(v_q) = \text{Beta}(v_q | \tau_{q,1}, \tau_{q,2}).$$

Then, the updates for parameters $\tau_{q,1}$ and $\tau_{q,2}$ are given by

$$\begin{aligned} \tau_{q,1} &= \alpha + \sum_{k=q+1}^{Q_+} \left[\sum_{d=1}^D (1 - \mathbb{E}[Z_{d,k}]) \sum_{j=q+1}^k q_{k,j} \right] + \sum_{k=q}^{Q_+} \sum_{d=1}^D \mathbb{E}[Z_{d,k}], \\ \tau_{q,2} &= 1 + \sum_{k=q}^{Q_+} \sum_{d=1}^D (1 - \mathbb{E}[Z_{d,k}]) q_{k,q}, \end{aligned}$$

For the evaluation of $p(\mathbf{Z}|\mathbf{v})$ in equation (3.6), we require to evaluate $\mathbb{E}[\log(1 - \prod_{i=1}^q v_i)]$, which has no closed-form solution. Hence, we resort to a local variational approximation ([Bishop, 2006](#)), where a multinomial distribution $q_q(y)$ bounds this expected value as

$$\begin{aligned} \mathbb{E}[\log(1 - \prod_{i=1}^q v_i)] &\geq \left(\sum_{m=1}^q q_{k,m} \psi(\tau_{m,2}) \right) + \left(\sum_{m=1}^{q-1} \left(\sum_{n=m+1}^q q_{q,n} \right) \psi(\tau_{m,1}) \right) \\ &\quad - \left(\sum_{m=1}^q \left(\sum_{n=m}^q q_{q,n} \right) \psi(\tau_{m,1} + \tau_{m,2}) \right) - \sum_{m=1}^q q_{q,m} \log(q_{q,m}), \end{aligned}$$

where $\psi(\cdot)$ is the digamma function. Besides, the update for the multinomial distribution is calculated as

$$q_{k,i} \propto \exp \left(\psi(\tau_{i,2}) + \sum_{m=1}^{i-1} \psi(\tau_{m,1}) - \sum_{m=1}^i \psi(\tau_{m,1} + \tau_{m,2}) \right).$$

3.3.2 Updates for $q(Z_{d,q})$

Inspired by [Doshi-Velez et al. \(2009\)](#), we also assume that each variational distribution for $Z_{d,q}$ is given by

$$q(Z_{d,q}) = \text{Bernoulli}(Z_{d,q} | \eta_{d,q}).$$

Thus, the updates for $\eta_{d,q}$ are calculated as

$$\eta_{d,q} = \frac{1}{1 + \exp(-\vartheta_{d,q})},$$

where $\vartheta_{d,q}$ is obtained from the canonical parametrization of the Bernoulli distribution and regarding the lower bound defined in equation (3.6), it takes the following form

$$\begin{aligned}\vartheta_{d,q} = & \text{tr} \left(\mathbf{m}_{d,q} \mathbb{E}[\mathbf{u}_q^\top] \right) - \sum_{q' \neq q} \eta_{d,q'} \text{tr} \left(\mathbf{P}_{d,q,q'} \mathbb{E}[\mathbf{u}_{q'} \mathbf{u}_q^\top] \right) \\ & - \frac{1}{2} \text{tr} \left(\mathbf{P}_{d,q,q} \mathbb{E}[\mathbf{u}_q \mathbf{u}_q^\top] \right) - \frac{c_{d,q}}{2} + \mathbb{E}[\log \pi_q] \\ & - \mathbb{E}[\log(1 - \pi_q)],\end{aligned}$$

with

$$\begin{aligned}\mathbf{P}_{d,q,q'} &= \beta_d \mathbf{K}_{\mathbf{u}_q, \mathbf{u}_q}^{-1} \boldsymbol{\Psi}_{d,q,q'} \mathbf{K}_{\mathbf{u}_{q'}, \mathbf{u}_{q'}}^{-1}, \\ c_{d,q} &= \beta_d \left(\xi_{d,q} - \text{tr} \left(\boldsymbol{\Psi}_{d,q,q} \mathbf{K}_{\mathbf{u}_q, \mathbf{u}_q}^{-1} \right) \right), \\ \mathbb{E}[\log \pi_q] &= \sum_{i=1}^q [\psi(\tau_{i,1}) - \psi(\tau_{i,1} + \tau_{i,2})], \\ \mathbf{m}_{d,q} &= \beta_d \mathbf{K}_{\mathbf{u}_q, \mathbf{u}_q}^{-1} \boldsymbol{\psi}_{d,q},\end{aligned}$$

where $\tau_{q,1}$ and $\tau_{q,2}$ are the parameters of the posterior $q(v_q)$.

3.3.3 Posterior distribution for the latent forces

Initially, we derive the form of the posterior $q(\mathbf{u})$ by collecting the terms related to \mathbf{u} from (3.6), as

$$\mathcal{F}_{\mathbf{u}} = \int q(\mathbf{u}) \left\{ \mathbb{E}_{p(\mathbf{F}|\mathbf{u})q(\mathbf{Z})} [\log(p(\mathbf{y}|\mathbf{F}, \mathbf{Z})p(\mathbf{u}))] - \log q(\mathbf{u}) \right\} d\mathbf{u}.$$

By merging the above logarithms and evaluating the expected value, the above expression becomes

$$\begin{aligned}\mathcal{F}_{\mathbf{u}} = & \int_{\mathbf{u}} q(\mathbf{u}) \log \left[\frac{\mathcal{N}(\mathbf{u}|\tilde{\mathbf{u}}, \tilde{\mathbf{K}}_{\mathbf{u},\mathbf{u}})}{q(\mathbf{u})} \right] d\mathbf{u} + \frac{1}{2} \tilde{\mathbf{u}}^\top \tilde{\mathbf{K}}_{\mathbf{u},\mathbf{u}}^{-1} \tilde{\mathbf{u}} + \\ & \frac{1}{2} \log |\tilde{\mathbf{K}}_{\mathbf{u},\mathbf{u}}| - \frac{1}{2} \log |\mathbf{K}_{\mathbf{u},\mathbf{u}}| - \frac{1}{2} \sum_{d=1}^D \sum_{q=1}^{Q_+} \mathbb{E}[Z_{d,q}] c_{d,q},\end{aligned}\tag{3.7}$$

where $\mathbf{K}_{\mathbf{u},\mathbf{u}} \in \mathbb{R}^{Q_+M \times Q_+M}$ is a block diagonal matrix with each block calculated using $k_{u_q, u_q}(t, t')$, and

$$\tilde{\mathbf{u}} = \tilde{\mathbf{K}}_{\mathbf{u},\mathbf{u}} \mathbf{m}, \quad \tilde{\mathbf{K}}_{\mathbf{u},\mathbf{u}}^{-1} = \mathbf{P} + \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1},\tag{3.8}$$

where $\mathbf{m} = [\mathbf{m}_1^\top, \dots, \mathbf{m}_{Q_+}^\top]^\top$, $\mathbf{m}_q = \sum_{d=1}^D \mathbb{E}[Z_{d,q}] \mathbf{m}_{d,q}$, and $\mathbf{P} \in \mathbb{R}^{Q_+M \times Q_+M}$ is a block-wise matrix with blocks given by

$$\mathbf{P}_{q,q'} = \mathbf{K}_{\mathbf{u}_q, \mathbf{u}_{q'}}^{-1} \sum_{d=1}^D \mathbb{E}[Z_{d,q} Z_{d,q'}] \beta_d \boldsymbol{\Psi}_{d,q,q'} \mathbf{K}_{\mathbf{u}_{q'}, \mathbf{u}_q}^{-1}.$$

Note that $\mathcal{F}_{\mathbf{u}}$ is maximized when the posterior

$$q(\mathbf{u}) = \mathcal{N}(\mathbf{u} | \tilde{\mathbf{u}}, \tilde{\mathbf{K}}_{\mathbf{u}, \mathbf{u}}),$$

and its moments are updated using the expressions defined in (3.8).

3.3.4 Hyperparameter learning

In order to learn the set of hyperparameters $\boldsymbol{\theta}$, we resort to use scaled conjugate gradient method. The idea is to search for a set of hyperparameters that maximises (3.6), while the parameters of the variational distributions are fixed.

Note that, as mentioned above, if the covariance function parameters $\boldsymbol{\theta}_{\text{kern}}$ are learned while the inducing variables \mathbf{u} are kept fixed, then, this slows down the learning process of both variables (Titsias and Lázaro-Gredilla, 2011). Fortunately, we are able to address this problem by marginalizing the latent forces from (3.6) using the result obtained in (3.7). Thus, the term for the lower bound used to learn the hyperparameters is

$$\begin{aligned} \mathcal{F}_{\boldsymbol{\theta}} = & \frac{1}{2} \tilde{\mathbf{u}}^\top \tilde{\mathbf{K}}_{\mathbf{u}, \mathbf{u}}^{-1} \tilde{\mathbf{u}} + \frac{1}{2} \log |\tilde{\mathbf{K}}_{\mathbf{u}, \mathbf{u}}| - \frac{1}{2} \log |\mathbf{K}_{\mathbf{u}, \mathbf{u}}| - \frac{1}{2} \sum_{d=1}^D \log |\boldsymbol{\Sigma}_{\mathbf{f}_d}| \\ & - \frac{1}{2} \sum_{d=1}^D \beta_d \mathbf{y}_d^\top \mathbf{y}_d - \frac{1}{2} \sum_{d=1}^D \sum_{q=1}^Q \mathbb{E}[Z_{d,q}] c_{d,q}. \end{aligned} \quad (3.9)$$

The variational EM algorithm adopted to find the number of latent forces is summarized in Algorithm 3. Note that the E-step is comprised in the iterative process defined in lines 4-8 of Algorithm 3. Additionally, the M-step is defined in line 9. At any iteration of both steps, we require $O(N(MQ_+)^2)$ time in order to update the posterior of \mathbf{u} and to evaluate the lower bound defined in (3.9). Thus, the computational burden is controlled by the truncation level Q_+ and the number of inducing points M , and it does not exponentially scale with the number of data points N .

Algorithm 3 Model selection in Latent force models based on the Indian Buffet Process.

- 1: **Input:** Training data (\mathbf{t}, \mathbf{y}) , truncation level Q_+ and α .
 - 2: Initialize hyper-parameters $\boldsymbol{\theta}$, and variational factors $\mathbf{v}, \mathbf{Z}, \mathbf{u}$.
 - 3: **while** (3.6) has not converged **do**
 - 4: **repeat**
 - 5: Update \mathbf{u} as described in section 3.3.3.
 - 6: Update \mathbf{v} as described in section 3.3.1.
 - 7: Update \mathbf{Z} as described in section 3.3.2.
 - 8: **until** (3.6) has converged.
 - 9: Update $\boldsymbol{\theta}$ by maximizing (3.9).
 - 10: **end while**
 - 11: **Return:** $\boldsymbol{\theta}, q(\mathbf{v}), q(\mathbf{Z}), q(\mathbf{u})$
-

3.4 Predictive distribution

Let us assume we are interested in predicting the output values $\mathbf{y}^* = [\mathbf{y}_1^{*\top}, \dots, \mathbf{y}_D^{*\top}]$ at the test time stamps $\mathbf{t}^* = [\mathbf{t}_1^{*\top}, \dots, \mathbf{t}_D^{*\top}]$, where each \mathbf{y}_d^* and $\mathbf{t}_d^* \in \mathbb{R}^{N_d^*}$, with N_d^* being the number of test points at the d -th output. Thus, we define the predictive distribution as

$$p(\mathbf{y}^*|\mathbf{y}) = \int_{\mathbf{Z}} \int_{\mathbf{F}^*, \mathbf{u}} p(\mathbf{y}^*|\mathbf{F}^*, \mathbf{Z}) p(\mathbf{F}^*|\mathbf{u},) q(\mathbf{u}) q(\mathbf{Z}) d\mathbf{F}^* d\mathbf{u} d\mathbf{Z}.$$

We can straightforwardly marginalize the distributions that are based on normal distributions (i.e. \mathbf{F}^* and \mathbf{u}), and obtain the following expression,

$$\begin{aligned} q(\mathbf{y}^*|\mathbf{Z}) &= \int_{\mathbf{F}^*, \mathbf{u}} p(\mathbf{y}^*|\mathbf{F}^*, \mathbf{Z}) p(\mathbf{F}^*|\mathbf{u}) q(\mathbf{u}) d\mathbf{F}^* d\mathbf{u}, \\ &= \mathcal{N}(\mathbf{y}^* | \bar{\mathbf{K}}_{\mathbf{f}^*, \mathbf{u}} \mathbf{K}_{\mathbf{u}, \mathbf{u}}^{-1} \tilde{\mathbf{u}}, \mathbf{C}), \end{aligned}$$

with $\mathbf{C} = \bar{\mathbf{K}}_{\mathbf{f}^*, \mathbf{f}^*} - \bar{\mathbf{K}}_{\mathbf{f}^*, \mathbf{u}} \boldsymbol{\Gamma}_{\mathbf{u}, \mathbf{u}} \bar{\mathbf{K}}_{\mathbf{u}, \mathbf{f}^*} + \boldsymbol{\Sigma}_{\mathbf{f}^*}$, where $\boldsymbol{\Gamma}_{\mathbf{u}, \mathbf{u}} = \mathbf{K}_{\mathbf{u}, \mathbf{u}}^{-1} - \mathbf{K}_{\mathbf{u}, \mathbf{u}}^{-1} \tilde{\mathbf{K}}_{\mathbf{u}, \mathbf{u}} \mathbf{K}_{\mathbf{u}, \mathbf{u}}^{-1}$. $\bar{\mathbf{K}}_{\mathbf{f}^*, \mathbf{f}^*}$ is a $D \times D$ block-wise matrix, with elements of each block evaluated as $\sum_q Z_{d,q} k_{f_d, f_d}^{(q)}(t, t')$. $\bar{\mathbf{K}}_{\mathbf{f}^*, \mathbf{u}}$ is $D \times Q_+$ block-wise matrix, with elements of each block evaluated using $Z_{d,q} k_{f_d, u_q}(t, t')$. Thus, the predictive distribution reduces to

$$p(\mathbf{y}^*|\mathbf{y}) = \int_{\mathbf{Z}} q(\mathbf{y}^*|\mathbf{Z}) q(\mathbf{Z}) d\mathbf{Z}.$$

Unfortunately, the above integral is intractable because the $Z_{d,q}$ variable appears on the determinant of $q(\mathbf{y}^*|\mathbf{Z})$. Since we are only interested in the mean and variance of

$p(\mathbf{y}_d^*|\mathbf{y})$, we can approximate it as

$$\mathbb{E}[\mathbf{y}_d^*] = \sum_{q=1}^{Q_+} \mathbb{E}[Z_{d,q}] \mathbf{K}_{\mathbf{f}_d^*, \mathbf{u}_q} \mathbf{K}_{\mathbf{u}_q, \mathbf{u}_q}^{-1} \tilde{\mathbf{u}}_q,$$

and

$$\begin{aligned} \text{cov}[\mathbf{y}_d^*, \mathbf{y}_d^*] &= \sum_{q=1}^{Q_+} \mathbb{E}[Z_{d,q}] \left[\mathbf{K}_{\mathbf{f}_d^*, \mathbf{f}_d^*}^{(q)} - \mathbf{K}_{\mathbf{f}_d^*, \mathbf{u}_q} \boldsymbol{\Gamma}_{\mathbf{u}_q, \mathbf{u}_q} \mathbf{K}_{\mathbf{u}_q, \mathbf{f}_d^*} \right] \\ &\quad - \sum_{q=1}^{Q_+} \sum_{q' \neq q}^{Q_+} \mathbb{E}[Z_{d,q} Z_{d,q'}] \mathbf{K}_{\mathbf{f}_d^*, \mathbf{u}_q} \boldsymbol{\Gamma}_{\mathbf{u}_q, \mathbf{u}_{q'}} \mathbf{K}_{\mathbf{u}_{q'}, \mathbf{f}_d^*} + \boldsymbol{\Sigma}_{f_d^*}. \end{aligned}$$

3.5 Related work

This section briefly reviews some approaches similar to the latent force multi-output Gaussian process construction, where the number of latent functions can be selected regarding the value of an objective function or by Bayesian estimation methods.

In [Chai et al. \(2009\)](#), robot inverse dynamics are modelled by a multi-task Gaussian Process with different numbers of shared latent functions. Once the models are learned, the number of latent functions is selected according to the model with the highest Bayesian information criterion value. Hence, this method requires that the experimenter manually sets the range of values of the number of latent functions to be tested. In contrast, we propose an approximate Bayesian automatic selection of the number of latent functions. Furthermore, our approach also allows to estimate the sparse interconnection between the outputs and latent functions.

In a closely related work [Titsias and Lázaro-Gredilla \(2011\)](#), the problem of model selection is approached using the spike and slab distribution as prior over the weight matrix of a linear combination of Gaussian processes latent functions. The inference step is performed using a variational approach. Besides, this model assumes a factorized posterior for the latent functions, which not only leads to a looser lower bound, but also to an expensive Maximization step, where an iterative procedure is required in order to learn each variational distribution of the latent functions.

Comparing the proposed approach with the work presented in [Guarnizo et al. \(2015\)](#), we make two major modifications. First, due to the strong dependency between the sensitivity values and the rest of the covariance function hyperparameters (i.e. the parameters involved in the ODE are highly correlated), we omitted the spike and slab

prior over the sensitivities values. Second, in [Guarnizo et al. \(2015\)](#), the posterior distribution for the latent functions is assumed to be independent across each latent force. On the other hand, in the proposed approach, this posterior is found from the distribution that maximizes the lower bound. These changes allow the proposed model to be less sensitive to the initialization of the parameters and the obtained lower bound is tighter.

3.6 Results

In this section, we show results from different datasets, including: synthetic, gene expression, weather and motion capture datasets. Our main focus is to find the number of latent functions required to adequately described the experiment's data. Nevertheless, we also include an analysis of the interconnection matrix \mathbf{Z} between the latent forces and outputs. For some results comparison, we adopt the normalised mean square error (NMSE) and the negative log probability density (NLPD), which are described in [appendix A](#).

The proposed variational method is carried out from 10 different initial conditions, among the trained models we selected the one that achieved the highest lower bound value. Thus, the results, analysed in the following subsections, are obtained from the above selected model. Furthermore, the estimated interconnection matrix is represented by the Hadamard product between the expected value of the IBP's variable $\mathbb{E}[\mathbf{Z}]$ and the estimated sensitivities $\hat{\mathbf{S}}$. This allow us to have a better picture of how the latent forces contribute to explain the data of each output. Additionally, only the values of $\mathbb{E}[\mathbf{Z}]$ larger than $1e^{-2}$ are considered. Thus, if all the values of a column of $\mathbb{E}[\mathbf{Z}]$ do not fulfil this constraint, then this column is removed. Finally, note that the number of columns that fulfils the above constraint is considered as the estimated number of latent forces.

3.6.1 Synthetic data

To show the ability of the proposed method to recover the underlying structure between the output data and the latent forces, we apply the method to a toy multi-output dataset. Toy data is generated from the model explained in [section 3.2](#), with $D = 3$, $Q = 2$ and $\alpha = 1$. The covariance function used in this experiment is ODE2, explained in [Sect. 2.2.3](#), with spring values $B_1 = 4$, $B_2 = 1$ and $B_3 = 1$. Damper values C_1 , C_2 and C_3 are

set to 0.5, 4 and 1, respectively. Then we sample \mathbf{Z} from the IBP prior defined in (3.1). Thus, the sensitivities values used according to \mathbf{Z} are

$$\mathbf{Z} \odot \mathbf{S} = \begin{bmatrix} 1.34 & -0.25 \\ -1.65 & 0 \\ 0 & -0.52 \end{bmatrix},$$

where \odot is the Hadamard product (element-wise product). Note that the entries of $\mathbf{Z} \odot \mathbf{S}$ different from zero represent the entries of \mathbf{Z} equal to one. For the covariance functions $k_q(t, t')$ of the latent forces, we choose the length-scales as $l_1 = 0.2$ and $l_2 = 0.4$. Finally, 50 data points per output were generated by sampling the model defined in (3.2) using the above parametrization. This data is corrupted by adding Gaussian white noise with variance 0.05. Additionally, some data points are used for training and others for testing as described in Table 3.1. For the variational approach procedure we assume that the level of truncation is $Q_+ = 4$ and $\alpha = 1$. Next, we proceed with the inference process as described in algorithm 3.

Table 3.1: Description of the number of data points used for training and validation for the toy experiment 3.6.1.

#	Training	Test
1	36	14
2	40	10
3	50	0

Figure 3.1 shows a comparison between the true Hinton diagram and the one obtained using the proposed approach. Although the number of latent forces is well estimated, the interconnection matrix presents some differences. Note that the first latent force, estimated by the proposed approach, is used to describe not only outputs 1 and 2 (as in the generative process), but also a small gain response from this latent force is used to explain the data of output 3. In consequence, the estimated first latent force function is broadly similar to the true one, except after time 3.5 s where the estimated function presents a wavy behaviour, as shown in figure 3.2.

On the other hand, the estimated sensitivity values associated to the second latent force have a contrary sign compared with their ground truth. Thus, the estimated second latent force is negative with respect to the true one, as shown in figure 3.2.

Next, we proceed to evaluate the performance of the model learned using the variational approach by predicting the test values of outputs one and two, as shown in table

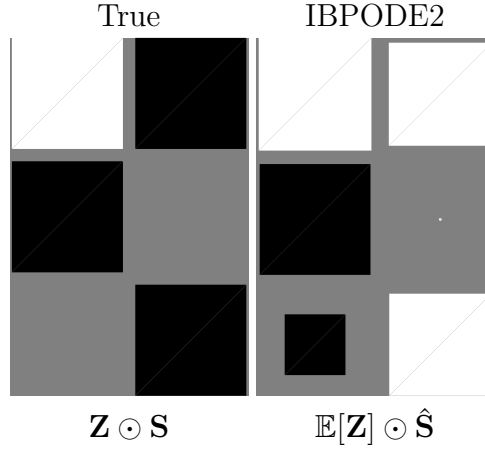


Figure 3.1: Hinton diagrams for the true network (left side) and the one obtained using the proposed variational approach (right side).

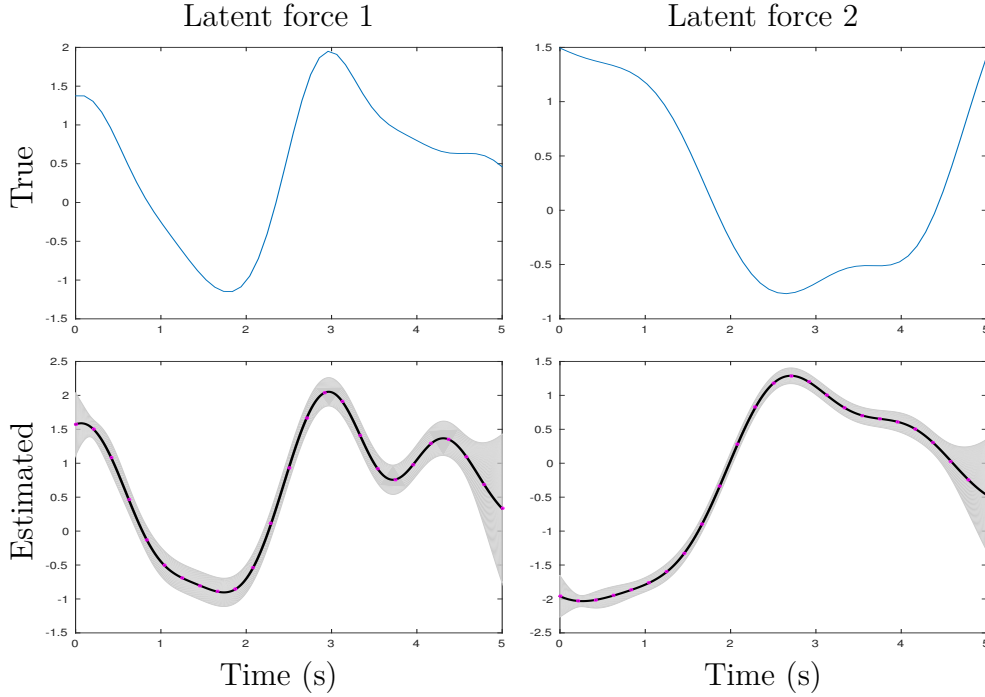


Figure 3.2: (Top) True waveform of the latent forces used to generate the toy data. (Bottom) Mean value (black line), two times standard deviation (grey) and inducing values (red dots) for the predictive GP of the latent forces estimated using the proposed approach.

3.2. Notice that the amplitude values for output two are in the range $[-0.6, 0.2]$, the noise added during the generation of the data affected considerably the amplitude values for this output. Thus, the NMSE value obtained for output two is larger than the one

Table 3.2: NMSE and NLPD measurements for testing data in toy experiment.

	NMSE	NLPD
Output 1	0.0013	-2.6281
Output 2	0.5071	-2.2373

obtained for output one. According to the NLPD values, test data for both outputs are well fitted by the model trained using the variational approach, as shown in figure 3.3.

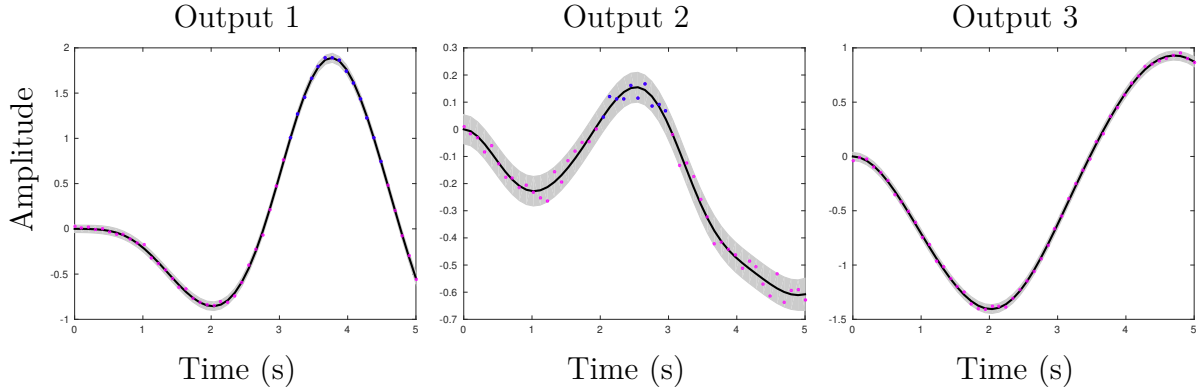


Figure 3.3: Mean value (black line), two times standard deviation (grey shadow), training (red dots) and test (blue dots) data for the predictive GP of outputs estimated using the proposed approach.

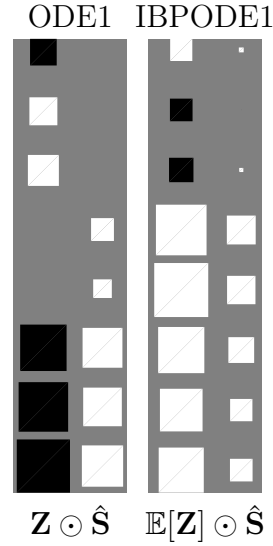
3.6.2 Yeast metabolic cycle data

For this experiment, we are interested in estimating the yeast transcriptional regulatory network from eight genes that comprises three cycles of yeast respiration measured using microarrays, as described in [Tu et al. \(2005\)](#). Each gene data consists of 36 time points sampled at 25 min intervals. The ribosomal production of each gene is regulated by two transcriptional factors (TFs), known as FHL1 and RAP1 as shown in Table 3.3. The regulatory network is known from Chip-on-chip data ([Oppen and Sanguinetti, 2010](#)) and is shown in the left side of figure 3.4.

In general, the dynamic relationship between gene expression and transcriptional regulation can be modelled by a first order ODE ([Lawrence et al., 2006](#)). Thus, in this experiment, the gene expression data is represented by the covariance function ODE1, which is described in Sect. 2.2.2. Additionally, we performed the variational procedures by assuming a level of truncation $Q_+ = 8$ and the number of inducing variables $M = 18$.

Table 3.3: Description of yeast data used in experiment 3.6.2.

#	Name	Regulated by
1	YLR183C	FHL1
2	YLR030W	FHL1
3	TKL2	FHL1
4	YOR359W	RAP1
5	PFK27	RAP1
6	RPL17B	FHL1,RAP1
7	RPS16B	FHL1,RAP1
8	RPL13A	FHL1,RAP1

Figure 3.4: Hinton diagrams obtained using the true \mathbf{Z} (left side) and the proposed variational approach (right side).

We compare the results of our proposed approach with a variational LFM ([Álvarez et al., 2009](#)) where the binary matrix \mathbf{Z} is assumed to have the form of the Chip-on-chip data. Hinton diagrams, for both approaches, are shown in figure 3.4. Note that the number of latent functions estimated by the proposed approach corresponds to the number of transcription factors involved in the regulation of the eight genes selected for this experiment. Additionally, the regulatory network given by the second latent force was inferred accurately from our proposed approach.

Figure 3.5 shows the latent forces estimated using the true \mathbf{Z} (first row) and the proposed approach (second row). The first latent force estimated by the proposed approach is broadly similar to both latent forces estimated by the variational LFM. Consequently,

in the proposed approach, the first latent force is used to describe the dynamics of all genes, as shown in figure 3.4. Meanwhile, the second latent force is used to describe the residual data over the last five genes.

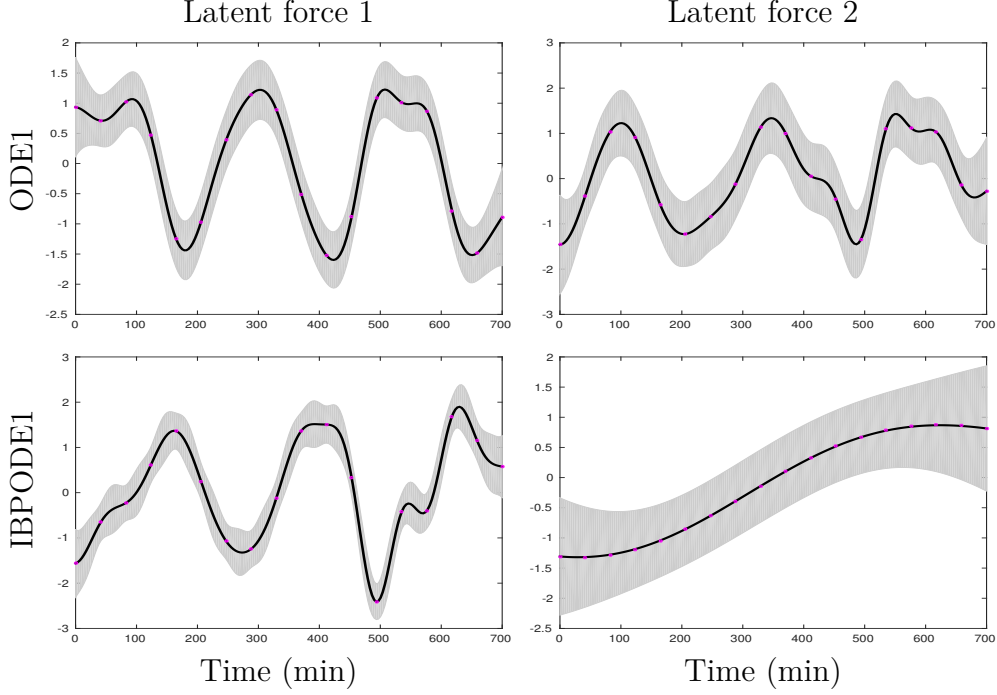


Figure 3.5: Mean value (black line), two times standard deviation (grey) and inducing values (red dots) for the predictive GP of the latent forces, which are estimated by assuming \mathbf{Z} known (first row) and the proposed approach (second row).

The reason that the expression data from the eight genes were mainly explained by one latent force is because the dynamic system modelled in equation (3.2) is linear. Hence, as stated in [Opper and Sanguinetti \(2010\)](#), if we are interested in obtaining an adequate approximation of the TFs, then non-linear dynamic effects must be considered. Nevertheless, if the system is assumed to be linear, then the data from the 8 genes is mainly driven by the TF RAP1. That is, our proposed approach estimated the TF RAP1 as the first latent force, and although the estimation of the second latent force is unrelated to TF FHL1 (due to there is no periodic behaviour), it contributes on the description of the genes regulated by this TF.

3.6.3 Human motion capture data

For this experiment, we consider the CMU motion capture (MOCAP) dataset, which consists of measured joint angles from different types of motions. We used the movement

“walking” from subject 02 motion 03. From the 62 available channels, we selected six that contained most of variability of the data along the x-axis. Additionally, data were downsampled by a factor of four. Table 3.4 summarizes the number of data points used for training and testing, and the names of the channels used for this experiment. Our

Table 3.4: Description of MOCAP angles data used in experiment 3.6.3.

#	Name	Training	Test
1	rhumeral	35	9
2	rradius	44	0
3	rfemur	44	0
4	lhumerus	35	9
5	lradius	44	0
6	lfemur	44	0

objective is to find the number of latent forces and how they contribute in the explanation of the MOCAP angles. With that in mind, we adopted the covariance function ODE2, which is adequate to model the motion of a human body.

We performed the variational procedures by assuming a level of truncation $Q_+ = 6$ and the number of inducing variables $M = 25$.

Figure 3.6 shows the Hinton diagrams obtained from the variational LFM and the proposed approach. We can see that the proposed variational approach found four latent

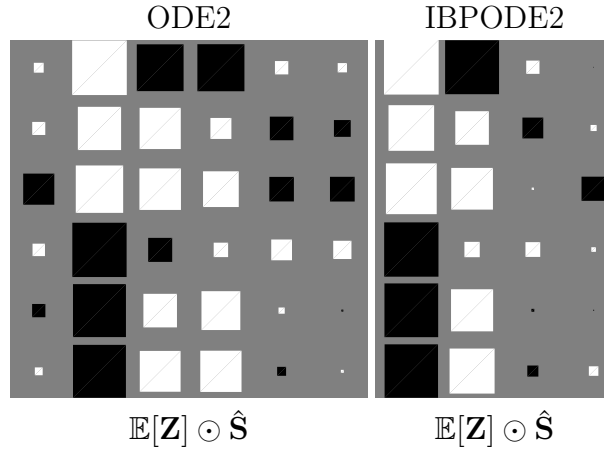


Figure 3.6: Hinton diagrams obtained assuming fixed full connectivity (left side) and using the proposed variational approach (right side).

forces, where the first two are shared among all outputs. Specifically, the first column is involved with the natural behaviour of walking cycle. That is, while a person is walking,

Table 3.5: NMSE and NLPD measurements for testing data in MOCAP experiment.

	ODE2		IBPODE2	
	NMSE	NLPD	NMSE	NLPD
rhumerus	0.0599	4.7309	0.2928	3.8969
lhumerus	0.0660	3.1147	0.0739	3.3267

if the left arm goes forward, then its right arm goes backward. Thus the first latent force acts positively for the right side channels and negatively for the left side channels.

The performance of the trained models according to the ability of predicting missing values is summarized in Table 3.4 and figure 3.7. The test data at “rhumerus” and “lhumerus” outputs is better fitted by the variational LFM. However, its predictive variance for “rhumerus” output is larger than the one obtained by the proposed approach. Thus, the predictive distribution obtained by the variational LFM is underconfident. For that reason, the NLPD value obtained by the proposed approach for rhumerus output is better than the one obtained by the variational LFM. For lhumerus output, both models behaved similarly.

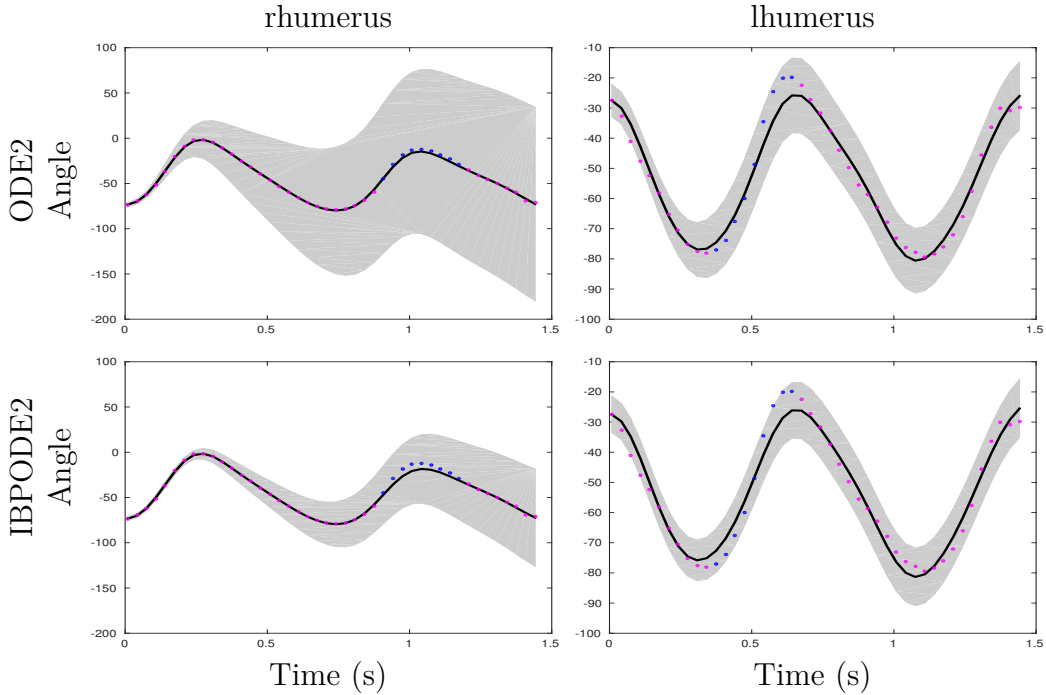


Figure 3.7: Mean value (black line), two times standard deviation (grey shadow), training (red dots) and test (blue dots) data for the predictive GP of outputs estimated using variational LFM (first row) and the proposed approach (second row).

3.6.4 Air temperature data

Here, we consider the problem of modelling and predicting air temperature time series from a network sensor located at the south coast of England. The dataset used in this experiment consists of the temperature measurements of four locations known as Bramblemet, Sotonmet, Cambermet and Chimet.¹ The air temperatures are measured during the period from July 10 to July 15, 2013. Furthermore, training and test data are arranged as proposed in [Nguyen and Bonilla \(2014\)](#) and described in Table 3.6.

Table 3.6: Description of Weather data used in experiment 3.6.4.

#	Name	Train	Test
1	Bramblemet	1425	0
2	Cambermet	1268	173
3	Chimet	1235	201
4	Sotonmet	1097	0

Note that temperature measurements at different locations may be correlated and hence we are able to make predictions about missing data using the available data from the other outputs at missing time stamps. Usually, the number of latent functions and how each latent function contribute to each output is assumed known a priori, as in [Álvarez and Lawrence \(2009\)](#); [Nguyen and Bonilla \(2014\)](#); [Osborne et al. \(2008\)](#). Interestingly, in [Nguyen and Bonilla \(2014\)](#) is proposed Collaborative Gaussian Process (CoGP) which is based on variational inducing points and assumes that each output is explained by a set of shared latent functions and an individual latent function. For comparison purposes, we adopted the same set-up used in [Nguyen and Bonilla \(2014\)](#), which consisted of two shared latent functions and 200 inducing points. The interconnection obtained, by the above parametrization, is presented as the left Hinton diagram in figure 3.8.²

For the proposed approach, we consider the Gaussian Smoothing covariance function (see section 2.2.4) and it is configured by setting the level of truncation $Q_+ = 6$ and the number of inducing variables $M = 200$, which is the same parametrization assumed by the CoGP approach. In the right side of figure 3.8 is shown the interconnection obtained by the best model found during the training phase. Note that the obtained model make use of the whole set of available latent functions. Furthermore, we can see that most of

¹Weather data can be found in <http://www.bramblemet.co.uk>.

²CoGP is coded in Matlab® and can be downloaded from <https://github.com/trungngv/cogp>.

the outputs are explained by 4 shared latent forces, meanwhile the outputs Chimet and Sotonet required an additional independent latent force.

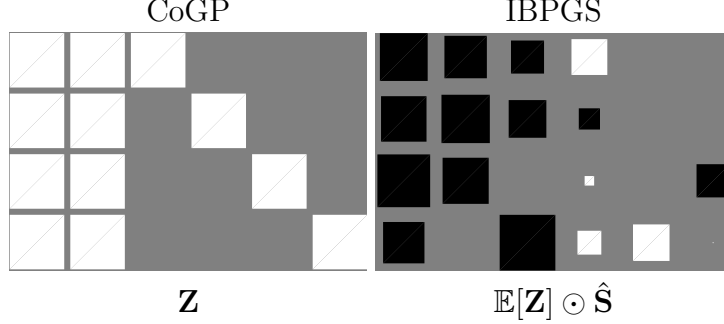


Figure 3.8: Hinton diagrams for the CoGP (left side) and using the proposed variational approach (right side).

Figure 3.9 shows the predictive distribution plots of Cambermet and Chimet outputs of models trained using CoGP and the proposed approach. For CoGP, we observe that not only the method poorly fits the training data, but also its variance is overconfident for the testing and training data. In contrast, the mean of our approach matches adequately the training data and also its variance covers the testing data. These results are corroborated by comparing the NMSE and NLPD measurements listed in Table 3.7, where our approach presented the best performance for predicting the testing data of each output.

Table 3.7: Comparison of IBPLFM and CoGP methods based on NMSE and NLPD measurements for testing data on the weather experiment.

	IBPGS		CoGP	
	NMSE	NLPD	NMSE	NLPD
Cambermet	0.1201	1.2490	0.1718	144.0671
Chimet	0.2975	1.1383	0.7417	97.6113

3.7 Discussion

We remark that estimating accurately both the interconnection matrix and the latent forces that generated a specific dataset is a complex task. First consider that the posteriors of the proposed approach may have multiple modes. Consequently, there are many different feasible solutions for the number of latent forces and the interconnection matrix

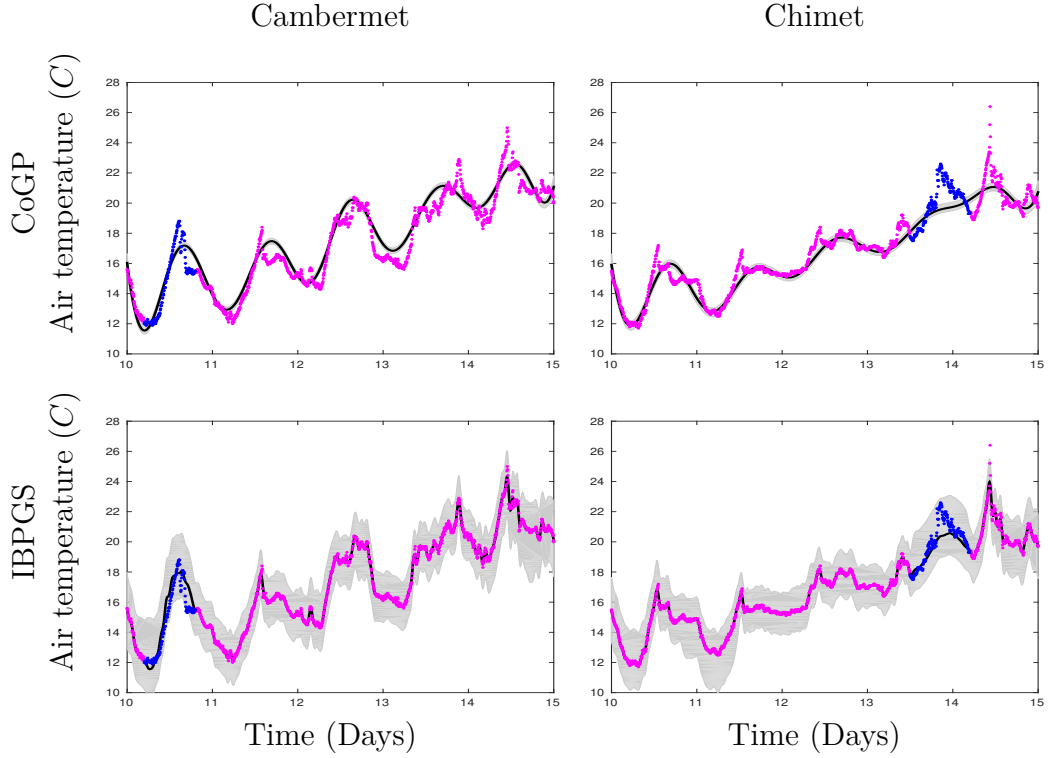


Figure 3.9: Mean value (black line), two times standard deviation (grey) and inducing values (red dots) for the predictive GP of the latent functions using the CoGP (first row) and the proposed approach (second row).

capable of describing the same data. Thus, in general, the parameters of the generative model are unidentifiable (Murphy, 2012, Chapter 11). However, from the results obtained in the toy and gene expression experiments, we are able to elucidate how the latent forces and \mathbf{Z} are constructed by the proposed approach. In both experiments, all outputs are mainly driven by the first latent force. In consequence, the residual data (remaining data that is not explained by the first latent force) is used to learn the next latent force function. The same process is followed in the estimation of the rest of the latent forces. Furthermore, this procedure for estimating the latent forces can be found in the sampled version of the IBP (Knowles and Ghahramani, 2011). Thus, we are unable to guarantee that the solution found by the proposed approach is precisely the parametrization used to generate the data.

Compared to the variational LFM and CoGP methods, the proposed approach is able to automatically learn both the number of latent functions and how these are interconnected to the outputs. Hence, if we assume a fixed interconnection in problems where there is no a priori knowledge about the relationship of the output data, this can

easily lead to an inadequate estimation of the latent functions. Nevertheless, the major drawback of the standard variational inference for the IBP prior is the requirement of a truncation level. Fortunately, this problem can be addressed by using an MCMC step in order to know if a new latent force is required to explain the remaining data at a specific output (Chatzis and Kosmopoulos, 2015; Knowles and Ghahramani, 2011). Still, in every MCMC step, we would require the optimization of the sensitivities and hyperparameters values related to the covariance function of the latent forces. Thus, we end up with a highly expensive process, which could be used or not regarding the acceptance of the MCMC step.

3.8 Conclusions

In this chapter a variational method based on LFMs and the IBP prior for inferring the number of latent forces in GP dynamic regression models is presented. We note that the IBP prior also allows to estimate the sparse interconnection between the outputs and the latent forces.

During the experiments, we found that the number of latent functions is accurately estimated over the toy data example. Interestingly, the proposed approach is able to partially estimate the gene regulatory network and the transcription factors solely from gene expression data.

The flexibility induced by the IBP prior over the LFM framework allows the proposed approach to avoid over/under-confident predictive distributions, as the ones obtained by the variational LFM and CoGP methods.

Chapter 4

Modelling multiple-input multiple-output data using Latent force models

In LFMs, we are able to estimate the excitation or input function $u(t)$ by assuming a partial knowledge about the dynamic linear system that models the data, i.e. we know the order of the differential equation which also lead us to know the parametric form of the Green's function or the impulse response function (IRF). In this chapter, we are instead interested on estimating the IRF from input and output data using the LFM approach.

IRF estimation tasks have been addressed using Orthogonal Basis Functions (OBFs) ([Reginato and Oliveira, 2007](#); [Stanislawski et al., 2008](#)). Laguerre functions are OBFs that are characterized by having only one pole or parameter that controls the waveform of the basis functions. Furthermore, they have been used in system identification tasks, as in [Israelsen and Smith \(2014\)](#); [Wahlberg \(1991\)](#). Hence, in this chapter, we propose to estimate the IRF using the Laguerre functions, which can be encoded in the covariance function of Convolved Gaussian Processes (CGPs). Laguerre parameters can be learned from the maximization of the CGP's marginal likelihood function.

This chapter is organized as follows. In section [4.1](#), Laguerre functions are introduced. From the Laguerre theory, we propose two models: Convolved Laguerre models and sequential Laguerre processes, which are described in sections [4.2](#) and [4.3](#), respectively. Additionally, some related works are reviewed in section [4.4](#). Some results showing the ability of the proposed models to estimate the IRFs under different scenarios are explored in section [4.5](#). Finally, some concluding remarks are discussed in section [4.6](#).

4.1 Laguerre functions

Laguerre functions form a complete and orthonormal set of basis. Each Laguerre function is defined as (Israelsen and Smith, 2014)

$$l_m(t) = \mathcal{L}_m(t) \exp(-\rho t),$$

where ρ is a free parameter known as the Laguerre scale (Haber and Keviczky, 1999), and

$$\mathcal{L}_m(t) = \sqrt{2\rho} \sum_{i=0}^m \frac{(-1)^i m! 2^{m-i}}{i! [(mi)!]^2} (2\rho t)^{m-i}.$$

The m -th degree polynomial $\mathcal{L}_m(t)$ is called the m -th Laguerre polynomial. It is also interesting to note that the Laguerre function $l_m(t)$ has m zero crossings defined by the zeros of $\mathcal{L}_m(t)$. Additionally, each Laguerre function represents a dynamical system characterized by one pole with multiplicity, e.g. the Laplace transform of the m -th Laguerre function is given by

$$L_m(s) = \sqrt{2\rho} \frac{(\rho - s)^m}{(\rho + s)^{m+1}},$$

where the Laguerre scale parameter ρ controls the position of the pole and the zeros of each Laguerre function. Hence, we are also using a dynamical representation to estimate the IRF.

4.2 Convolved Laguerre Process

The IRF of LFM is assumed to be known before hand in order to predict the input values (latent forces) from the output data. However, we are interested in approximating the IRF by means of Laguerre functions. In order to do so, we require to make use of input-output data. Thus, the IRF of the d -th output can be approximated by the Laguerre functions $l_{d,m}(t)$ as

$$G_d(t) \approx \sum_{m=0}^M c_{d,m} l_{d,m}(t), \quad (4.1)$$

where $c_{d,m}$ weights the m -th Laguerre function of the d -th output. Then, the convolution defined in (2.14) becomes

$$f_{d,q}(t) = \sum_{m=0}^M c_{d,m} \int_0^t l_{d,m}(t-\tau) u_q(\tau) d\tau. \quad (4.2)$$

Next, we proceed to define the covariance functions for the new model described in (4.2). Thus, the covariance function $k_{f_d, f_{d'}}^{(q)}(t, t')$ becomes

$$\int_0^t \sum_{m=0}^M c_{d,m} l_{d,m}(t-\tau) \int_0^{t'} \sum_{m=0}^M c_{d',m} l_{d',m}(t'-\tau') k_{u_q, u_q}(\tau, \tau') d\tau' d\tau, \quad (4.3)$$

and the cross covariance $k_{f_d, u_q}(t, t')$ is defined as

$$\sum_{m=0}^M c_{d,m} \int_0^t l_{d,m}(t-\tau) k_{u_q, u_q}(\tau, t') d\tau. \quad (4.4)$$

Note that, if $m > 0$, then the convolutions for the above covariance functions have no closed form. Consequently, we approximate the convolutions by using discrete sums as in Lawrence et al. (2006). This approximation increases the computation time for the evaluation of the covariance function, but it also allows the model to use any covariance function to model the inputs.

4.2.1 Hyperparameter learning

Let us assume we are given noisy observations of Q -inputs and D -outputs in vectors $\{\mathbf{v}_q\}_{q=1}^Q$ and $\{\mathbf{y}_d\}_{d=1}^D$, respectively. By conditioning the proposed GP model on this finite set of observations, the model becomes into the following multivariate normal distribution

$$\mathbf{z} = \begin{bmatrix} \mathbf{y} \\ \mathbf{v} \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{\mathbf{z}, \mathbf{z}}),$$

where \mathbf{z} is obtained by stacking in one column the observation vectors \mathbf{y} and \mathbf{v} . Furthermore, the covariance function for \mathbf{z} is defined as follows,

$$\mathbf{K}_{\mathbf{z}, \mathbf{z}} = \begin{bmatrix} \mathbf{K}_{\mathbf{f}, \mathbf{f}} + \Sigma_{\mathbf{f}} & \mathbf{K}_{\mathbf{f}, \mathbf{u}} \\ \mathbf{K}_{\mathbf{u}, \mathbf{f}} & \mathbf{K}_{\mathbf{u}, \mathbf{u}} + \Sigma_{\mathbf{u}} \end{bmatrix}, \quad (4.5)$$

where each $\mathbf{K}_{i,j}$ (i or j can be either \mathbf{f} or \mathbf{u}) matrix is obtained by evaluating the covariance function $k_{i,j}(t, t')$ at the time stamps \mathbf{t} associated to the observations \mathbf{z} . From the above definitions, the set of parameters $\boldsymbol{\theta}_{\text{CLP}} = \{l_q, \sigma_d^2, \rho_d, c_{d,m}\}_{d=1, q=1, m=0}^{D, Q, M}$ (where m indexes the Laguerre functions as described in (4.1)) are learned by maximizing the logarithm of the marginal likelihood (Rasmussen and Williams, 2006), which is given by

$$\underset{\boldsymbol{\theta}_{\text{CLP}}}{\text{maximize}} \quad \log p(\mathbf{z}|\mathbf{t}) = -\frac{1}{2}\mathbf{z}^\top \mathbf{K}_{\mathbf{z},\mathbf{z}}^{-1} \mathbf{z} - \frac{1}{2} \log |\mathbf{K}_{\mathbf{z},\mathbf{z}}| - \frac{N}{2} \log(2\pi),$$

where N is the total number of training data points represented by the length of vector \mathbf{z} .

4.2.2 Predictive distribution

According to (2.25), we are able to predict the values of the input and output functions \mathbf{z}^* at unknown time stamp values \mathbf{t}^* by using

$$\mathbf{z}^*|\mathbf{z} \sim \mathcal{N}\left(\mathbf{K}_{\mathbf{z}^*,\mathbf{z}}\mathbf{K}_{\mathbf{z},\mathbf{z}}^{-1}\mathbf{z}, \mathbf{K}_{\mathbf{z}^*,\mathbf{z}^*} - \mathbf{K}_{\mathbf{z}^*,\mathbf{z}}\mathbf{K}_{\mathbf{z},\mathbf{z}}^{-1}\mathbf{K}_{\mathbf{z},\mathbf{z}^*} + \boldsymbol{\Sigma}_{\mathbf{z}^*}\right), \quad (4.6)$$

where $\mathbf{K}_{\mathbf{z}^*,\mathbf{z}}$ is a $(D+Q) \times (D+Q)$ block-wise matrix with elements arranged in a similar form to (4.5), with the difference that rows are calculated using \mathbf{t}^* . Similarly, $\boldsymbol{\Sigma}_{\mathbf{z}^*}$ is $D+Q$ block diagonal matrix, where the elements of the j -th block are calculated by $\sigma_j^2 \mathbf{I}_{N_j^*}$, where N_j^* is the number of test points for the input or output located at that block.

4.3 Sequential Laguerre Processes

The model defined in (4.2) is converted into the following state-space representation,

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\epsilon(t), \quad \mathbf{z}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{r}(t), \quad (4.7)$$

where $\mathbf{z}(t) = [y_1(t), \dots, y_D(t), v_1(t), \dots, v_Q(t)]^\top$ comprises the noisy versions of the output and input functions, $\mathbf{x}(t) = [\mathbf{l}(t), \mathbf{u}(t)]^\top$ with $\mathbf{l}(t) = [l_{1,1}(t), \dots, l_{1,M}(t), l_{2,1}(t), \dots, l_{D,M}(t)]$ represents the Laguerre functions used along the outputs, $\mathbf{u}(t) = \left[u_1(t), \frac{du_1(t)}{dt}, \dots, \frac{d^{K-1}u_1(t)}{dt^{K-1}}, u_2(t), \dots, \frac{d^{K-1}u_Q(t)}{dt^{K-1}}\right]$ comprises the derivatives of all input functions, $\mathbf{r}(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$ models the noise process that corrupts the response and excitation functions,

with $\mathbf{R} = \text{diag}([\sigma_{f_1}^2, \dots, \sigma_{f_D}^2, \sigma_{u_1}^2, \dots, \sigma_{u_Q}^2])$, and $\boldsymbol{\epsilon}(t) = [\epsilon_1(t), \dots, \epsilon_Q(t)]^\top$ is the vector of white noise processes used by the SGP prior to model each input function $u_q(t)$ (for $q = 1, \dots, Q$). Matrices for the model described in (4.7) are given as

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_f & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_u \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} \mathbf{A}_f & \mathbf{B}_{fu} \\ \mathbf{0} & \mathbf{A}_u \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{0} \\ \mathbf{B}_u \end{bmatrix},$$

where $\mathbf{C}_f \in \mathbb{R}^{D \times DM}$ and $\mathbf{A}_f \in \mathbb{R}^{DM \times DM}$ are diagonal block-wise matrices of D -blocks with each block given by $\mathbf{C}_{f_d} = [c_{d,1}, \dots, c_{d,M}]$ and

$$\mathbf{A}_{f_d} = \begin{bmatrix} -\rho_d & 0 & \dots & 0 \\ -2\rho_d & -\rho_d & \dots & 0 \\ \vdots & \ddots & \ddots & \\ -2\rho_d & \dots & -2\rho_d & -\rho_d \end{bmatrix},$$

respectively. Also, $\mathbf{C}_u \in \mathbb{R}^{Q \times QK}$, $\mathbf{A}_u \in \mathbb{R}^{QK \times QK}$ and $\mathbf{B}_u \in \mathbb{R}^{QK \times Q}$ are diagonal block-wise matrices of Q -blocks with each block defined as $\mathbf{C}_{u_q} = [1, 0, \dots, 0]$ and

$$\mathbf{A}_{u_q} = \begin{bmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ & & 0 & 1 \\ -a_{q,0} & \dots & -a_{q,K-2} & -a_{q,K-1} \end{bmatrix}, \quad \mathbf{B}_{u_q} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}.$$

Coefficients $a_{q,i}$ (for $i = 0, \dots, K-1$) are used to approximate the Gaussian Process prior over the latent function $u_q(t)$, this GP is built from a Matérn covariance function (Hartikainen and Särkkä, 2010). Finally, $\mathbf{B}_{fu} \in \mathbb{R}^{DM \times QK}$ is a block-wise matrix where the block at the d -th row and q -th column is defined as

$$\mathbf{B}_{f_d u_q} = \begin{bmatrix} \sqrt{2\rho_d} & 0 & \dots & 0 \\ \sqrt{2\rho_d} & 0 & \dots & 0 \\ \vdots & & & \\ \sqrt{2\rho_d} & 0 & \dots & 0 \end{bmatrix}.$$

4.3.1 Sequential inference

We start by defining the following discrete observation model as

$$\mathbf{z}_k = \mathbf{C}\mathbf{x}_k + \mathbf{r}_k,$$

where $\mathbf{z}_k = [y_1(t_k), \dots, y_D(t_k), u_1(t_k), \dots, u_Q(t_k)]^\top$ represents the training data at discrete time t_k , and \mathbf{r}_k models the noise process over the response and excitation time series.

We are able to learn the model defined in (4.7) by using the KF procedure, described in algorithm 1. Furthermore, the hyperparameters are found by maximizing the following objective,

$$\underset{\boldsymbol{\theta}_{\text{SLP}}}{\text{maximize}} \quad \log p(\mathbf{y}|\mathbf{t}) = -\frac{1}{2} \sum_{k=1}^N \left[\log |2\pi\mathbf{S}_k| + \mathbf{v}_k^\top \mathbf{S}_k^{-1} \mathbf{v}_k \right], \quad (4.8)$$

where $\boldsymbol{\theta}_{\text{SLP}}$ comprises the hyperparameters required to describe the model defined in (4.7), and the noise variances.

4.3.2 Predictive distributions

We are able to predict the value $\mathbf{z}(t^*)$ by including the test time t^* in the KF steps. Hence, the moments of the prediction are given by

$$\mathbb{E}[\mathbf{y}(t^*)] = \mathbf{C}\mathbf{m}_{t^*}, \text{ and } \mathbf{V}[\mathbf{y}(t^*)] = \mathbf{S}_{t^*},$$

where \mathbf{m}_{t^*} is the mean vector of the state vector, and \mathbf{S}_{t^*} is the observation covariance matrix. Both matrices are evaluated at time t^* .

4.4 Related work

The IRF can be approximated in a non-parametric manner by placing a GP prior over this function, except in the LFM approach because of the product of two GPs is intractable. In Tobar et al. (2015) the Gaussian Process Convolution Model (GPCM) has been introduced. The GPCM is described as a continuous-time non-parametric window moving average process. One advantage of this model is that it can be considered in the frequency domain as well. In order to avoid the intractability mentioned above, the excitation function is assumed to be a white noise process.

Another non-parametric approximation is proposed in [Risuelo et al. \(2016\)](#), where a specific covariance matrix is built by using the stable spline kernel. In this method, a GP prior is placed over the IRF, but the model is only defined in discrete time.

The above mentioned methods require to approximate the posterior distribution of the IRF by using special algorithms, i.e. the GPCM requires a variational inference approach, meanwhile the work in [Risuelo et al. \(2016\)](#) an Expectation-Maximization algorithm is adopted. In contrast, for the model proposed here, the IRF is estimated using a set of orthogonal basis (Laguerre functions) which included a dynamical representation. Additionally, the proposed model is learned using the standard GP training method ([Rasmussen and Williams, 2006](#)).

4.5 Results

In this section, two different numerical problems are given to illustrate the properties of the proposed models. In both experiments, a grid of 500 points is used for the CLP approach to approximate the convolutions described in (4.3) and (4.4). For some results comparison, we adopt the normalised mean square error (NMSE) and the negative log probability density (NLPD), which are described in appendix A.

4.5.1 Approximation of the impulse response

In this experiment, we show the ability of the proposed models to estimate the impulse response function by means of the Laguerre functions. First, we generated 50 data points equally spaced along the range $[0, 4]$ s, by sampling $u(t)$ from the GP based on a square exponential covariance function with $l = 0.8$. Then, the output data \mathbf{f} is obtained by applying \mathbf{u} through a second order dynamical system characterized by the following IRF

$$G(t) = \frac{1}{\omega} \exp\left(-\frac{b_1 t}{2}\right) \sinh(\omega t),$$

with $b_0 = 1$, $b_1 = 4$ and $\omega = \sqrt{b_1^2 - 4b_0}/2$. From the data vectors \mathbf{f} and \mathbf{u} we proceed to learn the models CLP and SLP with $M = 10$ Laguerre basis by using the procedure described in section 4.2.1 and 4.3.1, respectively. The models are learned 10 times with different parameters initializations. Figure 4.1 shows the mean and two standard deviations calculated from the 10 IRFs learned for the proposed models. Additionally, the SMSE's mean value for CLP and SLP are 0.0919 ± 0.1048 and 0.4743 ± 0.9662 ,

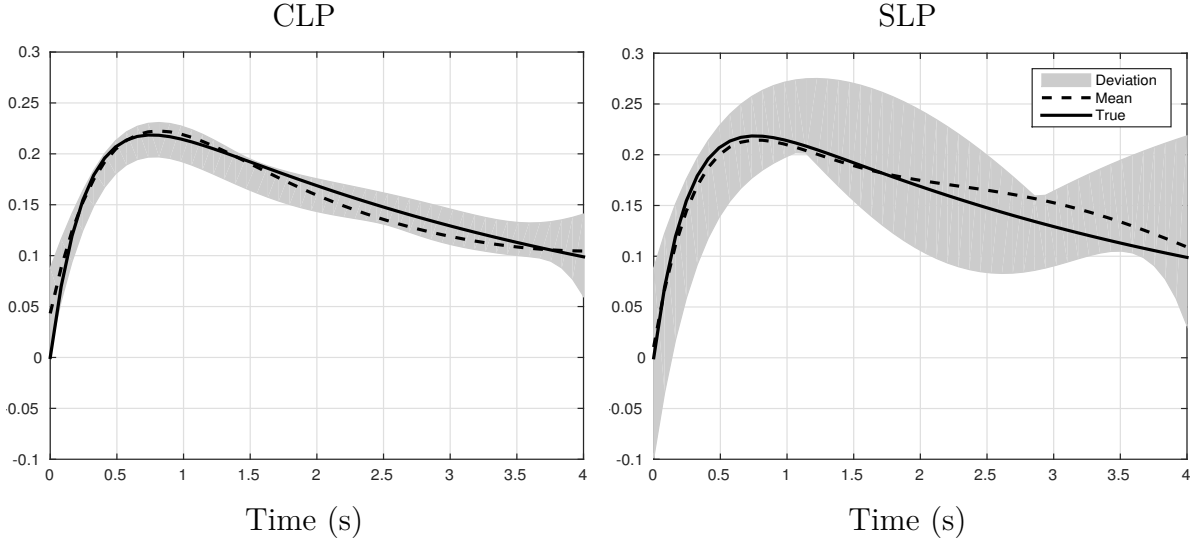


Figure 4.1: Mean and two standard deviations calculated from the impulse response functions estimated using CLP and SLP for experiment 4.5.1.

respectively. The mean of the SMSE values indicates that the CLP approach estimated better the IRF. Furthermore, according to figure 4.1 and the standard deviation of the SMSE values, the IRFs obtained by the CLP approach were the most accurate.

4.5.2 Prediction of missing input/output values

For this experiment, we are interested in predicting missing values of inputs and outputs. First, we configured a MIMO system with two inputs and two outputs described by the following equations

$$\frac{d^4 f_1(t)}{dt^4} + 4\frac{d^3 f_1(t)}{dt^3} + 9\frac{d^2 f_1(t)}{dt^2} + 14\frac{df_1(t)}{dt} + 8f_1(t) = \bar{u}(t),$$

$$\frac{df_2(t)}{dt} + f_2(t) = \bar{u}(t),$$

with $\bar{u}(t) = \sum_{q=1}^2 u_q(t)$. Then, 50 data points are generated for input and output variables (we follow similar steps as the ones described in experiment 4.5.1). Additionally, output data is corrupted by an additive white noise process. We proceed to learn both proposed models by assuming that the number of Laguerre functions is $M = 10$. Furthermore, in order to show the ability of the proposed models to deal with missing data, the dataset is divided into 2 subsets (training and testing) as shown in figure 4.2.

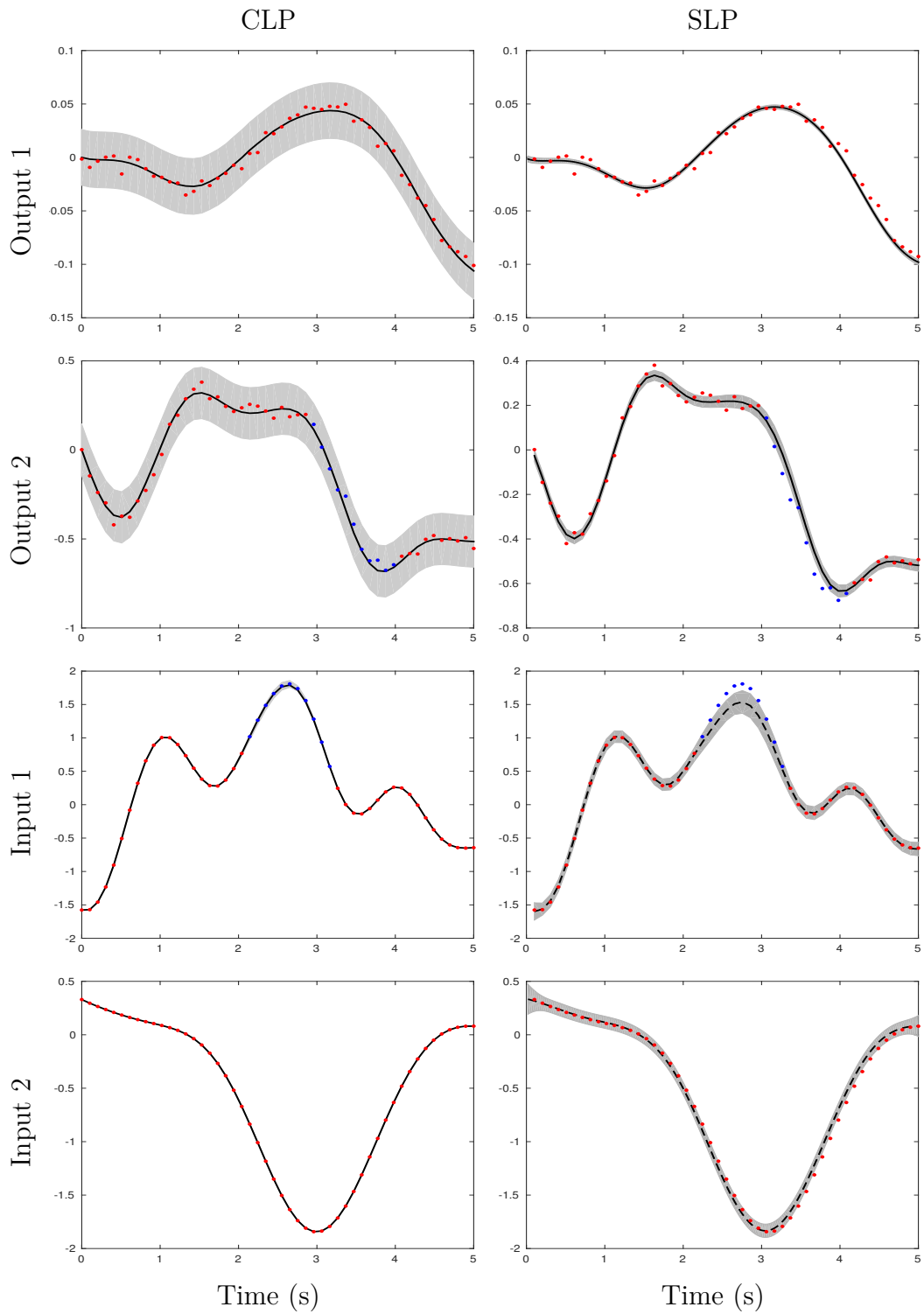


Figure 4.2: Prediction of missing data (Test data) for the MIMO system described in section 4.5.2. Predictive distributions are represented by the mean value (black line) and two times the standard deviation (grey shadow), training (red dots) and test (blue dots) data.

According to figure 4.2 and table 4.1, the predictive distributions found by the CLP approach are better suited to describe the testing data. Unfortunately, the SLP approach is overconfident to explain the output data, and the missing data from input 1 is not fitted adequately. The sequential inference for the SLP approach can behave poorly for consecutive testing points. In contrast, using all the observations (input and output values) simultaneously allow the CLP approach to better describe consecutive testing points.

Table 4.1: Comparison of CLP and SLP methods based on NMSE and NLPD measurements for the testing data of experiment 4.5.2.

	CLP		SLP	
	NMSE	NLPD	NMSE	NLPD
Output 2	0.0168	-1.5804	0.0521	2.6275
Input 1	0.0011	-3.104	0.2822	1.3418

The IRFs estimated for both outputs using the CLP and SLP approaches are shown in figure 4.3. We can see that the IRFs estimated from both approaches were affected by the missing data. Nevertheless, both models adequately matched the dynamic behaviour of the IRF for output 1, even for the small amplitude values ranging from -0.02 to 0.08.

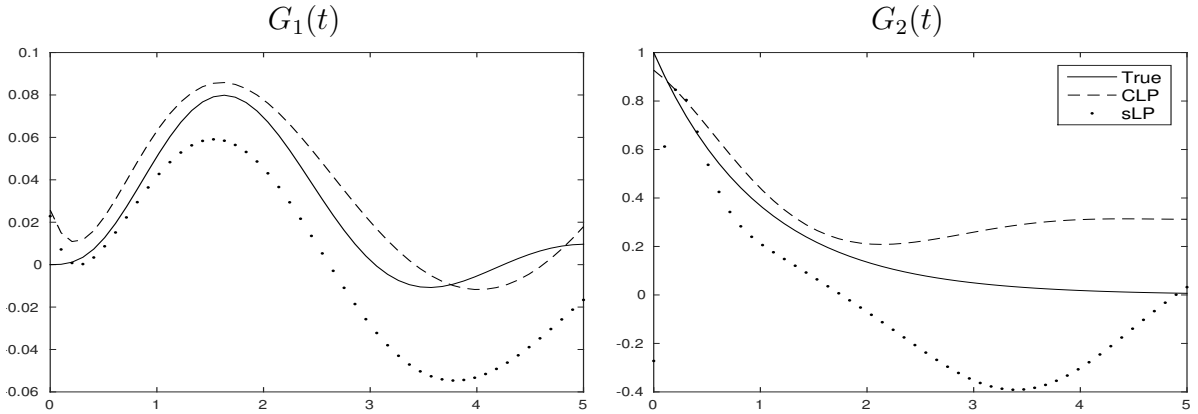


Figure 4.3: Impulse response approximation for the MIMO system described in section 4.5.2, using the proposed methods CLP and SLP.

4.5.3 CD-player arm

In this section, we consider the experimental data from a mechanical construction of a CD-player arm. The system consists of two inputs that are the forces of the mechanical

actuators, and two outputs involved in the tracking accuracy of the arm. The data was measured in closed loop, but through a two-step procedure it was converted to open loop equivalent data (De Moor et al., 1997). The data set contains 2048 sample points for each input or output. From the 2048 data points, we downsampled by 2 the first 400 data points. Thus, the selected 200 data points are arranged for training and testing as described in Table 4.2 and figure 4.4.

Table 4.2: Description of CD-player arm data used in experiment 4.5.3.

Name	Training	Test
Output 1	200	0
Output 2	189	11
Input 1	189	11
Input 2	200	0

The performance comparison between the CLP and SLP approaches related to the ability to explain the missing or test data is summarized in figure 4.4 and Table 4.3. Interestingly, the SLP approach better explained the missing data for Input 1, while the CLP approach performed best to describe the test data for Output 2. Additionally, note that the CLP approach performed best to describe the test data variability of Input 1 and Output 2 (regarding the NLPD measurements).

Table 4.3: Comparison of CLP and SLP methods based on NMSE and NLPD measurements for the testing data of CD-player arm experiment.

	CLP		SLP	
	NMSE	NLPD	NMSE	NLPD
Output 2	0.1350	-0.7355	0.3977	6.3932
Input 1	0.4433	-0.7062	0.1144	-1.3277

Figure 4.5 shows the estimated IRFs using the proposed approaches. The IRFs for Output 1 estimated by both approaches are similar in shape (they are scaled differently). However, the IRFs estimated for Output 2 differ in shape and magnitude, because of the uncertainty induced by the missing data.

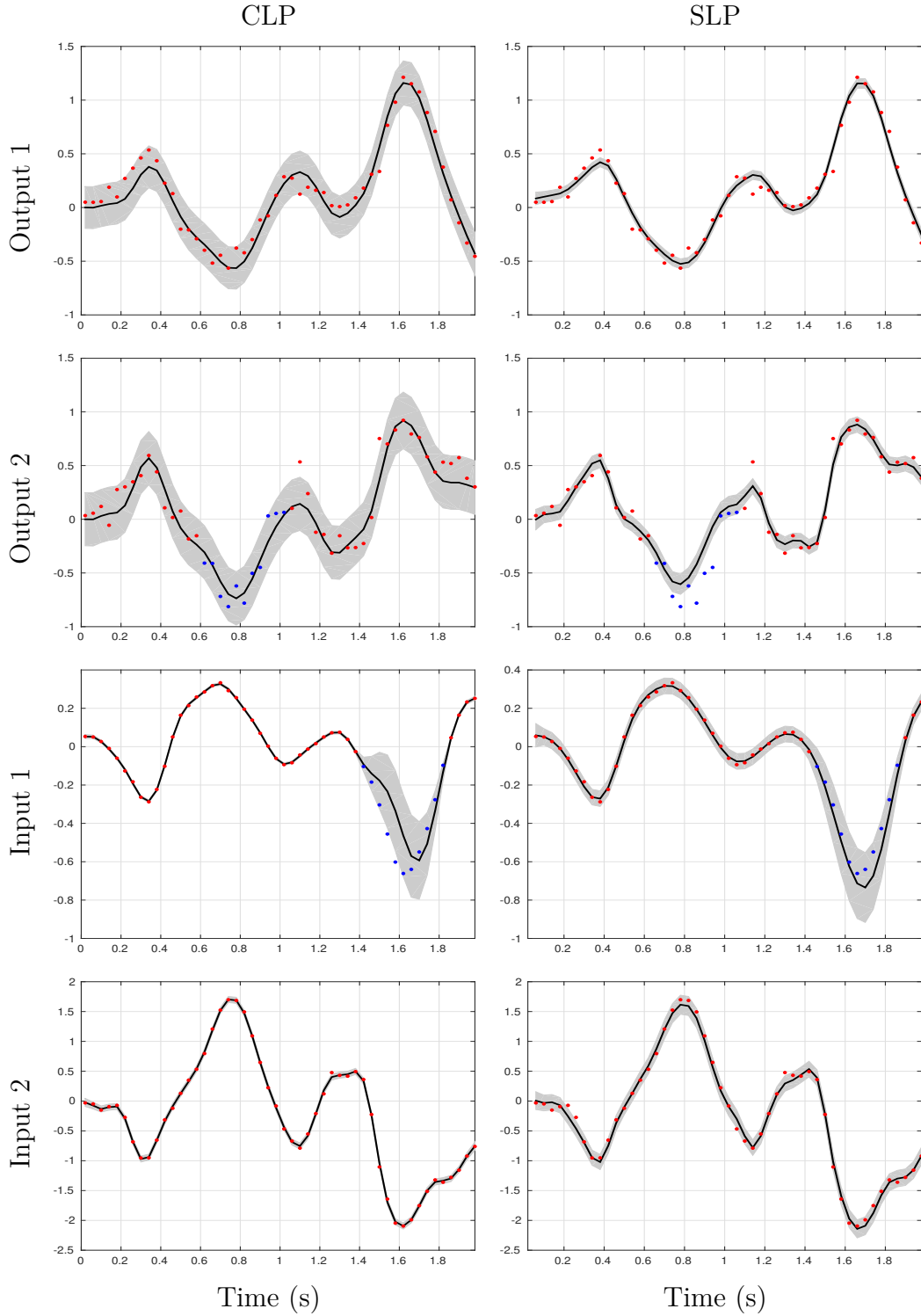


Figure 4.4: Prediction of missing data (Test data) for the CD-player arm system described in section 4.5.3. Predictive distributions are represented by the mean value (black line) and two times the standard deviation (grey shadow), training (red dots) and test (blue dots) data.

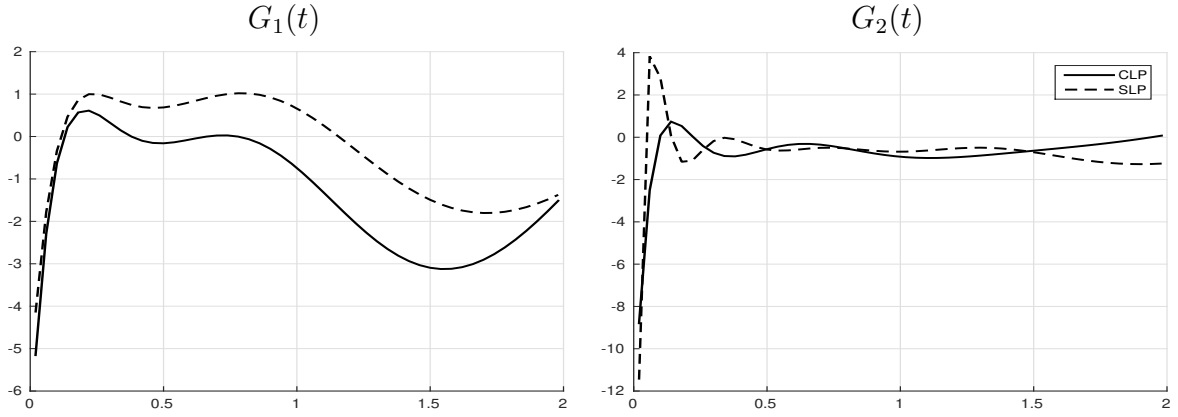


Figure 4.5: Impulse response approximation for the CD-player arm system described in section 4.5.3, using the proposed methods CLP and SLP.

4.6 Conclusions

In this chapter, we have proposed two different approaches aimed to model multiple-input and multiple-output data of LTI systems. Additionally, we are able to point-estimate the IRFs of each output from the hyper-parameters selected during the learning procedure. The first approach is based on the convolution construction of LFMs, while the second approach uses the state-space representation of LFMs.

The experiments demonstrated that the proposed approaches are able to model MIMO data and estimate the IRFs of LTI systems in the presence of noise corruption and missing input/output data. Furthermore, the CLP approach performed better, mostly because it uses all the available data at the prediction stage. However, the inference procedure of the CLP approach is highly expensive compared to the sequential procedure used by the SLP approach.

These models can be easily extended by using another set of orthonormal basis or dynamic-related functions. For example, selecting a set of basis which allows to find a closed form for the CLP's convolutions is recommended.

Chapter 5

Wiener system approximation using latent force models

Non-linear dynamical systems are, in general, better suited to described real word problems than their linear counterparts. In this chapter we focus on Wiener systems that consist on a non-linear static function applied over the response of a linear dynamical system as shown in figure 5.1. Specifically, we present two novel approaches that ap-

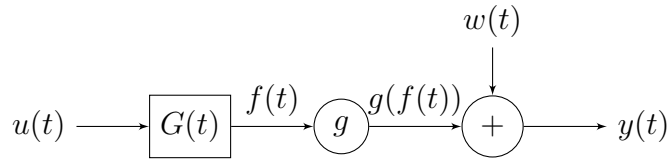


Figure 5.1: Block representation of a Wiener system.

proximate Wiener systems. The first approach is based on LFMs, and it is focused on estimating the excitation or input function from the output data. On the other hand, the second approach is based on SLMFs, and it is aimed to estimate the impulse response function using Laguerre functions (as in Chapter 4). In both approaches the non-linear static function, $g(\cdot)$, is approximated using linearisation methods, as described next.

5.1 Linearisation Methods

We are able to approximate the values of the non-linear transformation using the following relations:

$$g(\mathbf{f}(t)) \approx \mathbf{A}\mathbf{f}(t) + \mathbf{b}, \quad \mathbf{f}(t) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K}), \quad (5.1)$$

with $\boldsymbol{\mu}$ and \mathbf{K} being the mean vector and covariance matrix of $\mathbf{f}(t) \in \mathbb{R}^D$, respectively. Also, $\mathbf{b} \in \mathbb{R}^D$ and $\mathbf{A} \in \mathbb{R}^{D \times D}$. In consequence, $g(\mathbf{f}(t)) \in \mathbb{R}^D$ is a column vector with elements given by $g(f_d(t))$, for $d = 1, \dots, D$. In order to calculate \mathbf{A} and \mathbf{b} , we adopt the Taylor series and Statical linearisation methods, which are described next.

First order Taylor series

The approximation given by a first order Taylor series applied over the non-linear transformation of a Gaussian random variable, as in (5.1), can be represented as

$$g(\mathbf{f}(t)) \approx g(\boldsymbol{\mu}) + \mathbf{J}(\mathbf{f}(t) - \boldsymbol{\mu}),$$

with

$$\mathbf{J} = \left. \frac{\partial g(\mathbf{f}(t))}{\partial \mathbf{f}(t)} \right|_{\mathbf{f}(t)=\boldsymbol{\mu}}.$$

Hence, we are able to calculate (5.1) using

$$\mathbf{A} = \mathbf{J}, \quad \mathbf{b} = g(\boldsymbol{\mu}) - \mathbf{J}\boldsymbol{\mu}.$$

Statistical linearisation

Statistical linearisation allows us to find the least squares best fit of $g(f(t))$ around the point $f(t)$. To do so, we require to evaluate the static non-linear function at multiple points. These points are selected according to the unscented transform, as in [Dezfouli and Bonilla \(2015\)](#), which defines $2D + 1$ sigma points,

$$\begin{aligned} \mathcal{M}_0 &= \boldsymbol{\mu} \\ \mathcal{M}_i &= \boldsymbol{\mu} + \left(\sqrt{(D + \kappa)\mathbf{K}} \right)_i \quad \text{with } 1 \leq i \leq D \\ \mathcal{M}_i &= \boldsymbol{\mu} - \left(\sqrt{(D + \kappa)\mathbf{K}} \right)_i \quad \text{with } D < i \leq 2D \\ \mathcal{Y}_i &= g(\mathcal{M}_i) \quad \text{with } 0 \leq i \leq 2D, \end{aligned}$$

where κ is a free parameter, and $(\sqrt{\cdot})_i$ represents the i -th column of the matrix square root, which can be calculated using the Cholesky decomposition. From the relations obtained above we define the following statistics,

$$\bar{\mathbf{y}} = \sum_{i=0}^{2D} c_i \mathcal{Y}_i, \quad \boldsymbol{\Gamma} = \sum_{i=0}^{2D} c_i (\mathcal{Y}_i - \bar{\mathbf{y}}) (\mathcal{M}_i - \boldsymbol{\mu}_{\mathbf{f}})^\top, \quad (5.2)$$

with weight coefficients defined as

$$c_0 = \frac{\kappa}{D + \kappa}, \quad c_i = \frac{1}{2(D + \kappa)} \quad \text{with} \quad 0 < i \leq 2D.$$

We are able to find the requirements of (5.1) by solving the following objective,

$$\arg \min_{\mathbf{A}, \mathbf{b}} \sum_{i=0}^{2D} \|\mathcal{Y}_i - (\mathbf{A}\mathcal{M}_i + \mathbf{b})\|_2^2,$$

which corresponds to a linear least-squares problem with solution given by

$$\mathbf{b} = \bar{\mathbf{y}} - \mathbf{A}\boldsymbol{\mu}, \quad \mathbf{A} = \mathbf{\Gamma}\mathbf{K}^{-1}.$$

It is worth to mention that unlike Taylor series, the statistical linearisation method does not require derivative evaluations of the static non-linear function.

5.2 Latent force models for Wiener systems

As shown in figure 5.1, Wiener systems can be modelled as

$$y(t) = g(f(t)) + w(t), \tag{5.3}$$

where $f(t)$ follows a LFM prior, $g(\cdot)$ is an arbitrary warping function with scalar input, and $w(t)$ is a white noise process with variance σ_g^2 . In order to avoid the intractability induced by the static non-linear function, we adopt the extended and unscented GP approach proposed in Steinberg and Bonilla (2014). We start by assuming that we are given a dataset consisting of N noisy observed values, $\mathbf{y} \in \mathbb{R}^N$, which are obtained from the transformation of the response function $\mathbf{f} \in \mathbb{R}^N$. Specifically, if we condition the model given in (5.3) on the dataset, then it can be described using the following likelihood and prior,

$$\mathbf{y} \sim \mathcal{N}(g(\mathbf{f}), \sigma_g^2 \mathbf{I}_N), \quad \mathbf{f} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{\mathbf{f}, \mathbf{f}}). \tag{5.4}$$

Notice that we are unable to obtain the posterior $p(\mathbf{f}|\mathbf{y})$ because of the non-linear transformation of \mathbf{f} . Fortunately, we are able to address this problem by using variational inference (Bishop, 2006), as described next.

5.2.1 Inference

To overcome the problem imposed by the non-linear function in (5.4), we assume that the posterior of \mathbf{f} takes the form, $q(\mathbf{f}) = \mathcal{N}(\mathbf{f}|\mathbf{m}, \mathbf{C})$, where $\mathbf{m} \in \mathbb{R}^N$ is the posterior mean, and $\mathbf{C} \in \mathbb{R}^{N \times N}$ is the posterior covariance matrix. Thus, we can place a lower bound on the log-marginal likelihood as

$$\log p(\mathbf{y}) \geq \int q(\mathbf{f}) \log \frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f})}{q(\mathbf{f})} d\mathbf{f}.$$

Similarly to section 3.3, we are able to minimize the Kullback-Leibler distance between $p(\mathbf{f}|\mathbf{y})$ and $q(\mathbf{f})$ by maximizing the above lower bound, which becomes

$$\mathcal{F} = \mathbb{E}_{q(\mathbf{f})} [p(\mathbf{y}|\mathbf{f})] - \text{KL} [q(\mathbf{f})||p(\mathbf{f})], \quad (5.5)$$

where $\text{KL}(q||p)$ is the Kullback-Leibler distance between the distributions q and p . Given that the posterior is assumed to be Gaussian distributed, we can evaluate the expectation and KL term from the above equation as

$$\mathbb{E}_{q(\mathbf{f})} [p(\mathbf{y}|\mathbf{f})] = -\frac{1}{2} \left[N \log 2\pi\sigma_g^2 + \frac{1}{\sigma_g^2} \sum_{n=1}^N \mathbb{E}_{q(\mathbf{f})} [(y_n - g(f_n))^2] \right], \quad (5.6)$$

$$\text{KL} [q(\mathbf{f})||p(\mathbf{f})] = \frac{1}{2} \left[\text{tr}(\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1}\mathbf{C}) + \mathbf{m}^\top \mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1} \mathbf{m} - \log|\mathbf{C}| + \log|\mathbf{K}_{\mathbf{f},\mathbf{f}}| - N \right].$$

Note that we are able to evaluate the expectation in (5.6) by approximating the static non-linear function using the linearisation methods described in section 5.1. Besides, in this case, the linearisation is one-dimensional and it is given by

$$g(f_n) \approx a_n f_n + b_n,$$

where the coefficients a_n 's and b_n 's are calculated using the linearisation methods described in 5.1. Hence, we can approximate the expectation in (5.6) as,

$$\mathbb{E}_{q(\mathbf{f})} [p(\mathbf{y}|\mathbf{f})] \approx -\frac{1}{2} N \log 2\pi\sigma_g^2 - \frac{1}{2\sigma_g^2} \left[\mathbf{e}^\top \mathbf{e} + \text{tr}(\mathbf{A}^\top \mathbf{A} \mathbf{C}) \right], \quad (5.7)$$

with $\mathbf{e} = \mathbf{y} - (\mathbf{A}\mathbf{m} + \mathbf{b})$ and $\mathbf{A} = \text{diag}([a_1, \dots, a_N])$.

Learning the Variational moments

As in [Bonilla et al. \(2016\)](#); [Steinberg and Bonilla \(2014\)](#), given that there is no close form to update the posterior mean \mathbf{m} , we resort to Newton's method to find the approximate posterior mean,

$$\mathbf{m}^{(k+1)} = \mathbf{m}^{(k)} - \alpha \left(\frac{\partial^2 \mathcal{F}}{\partial \mathbf{m} \partial \mathbf{m}^\top} \right)^{-1} \frac{\partial \mathcal{F}}{\partial \mathbf{m}} \Big|_{\mathbf{m}=\mathbf{m}^{(k)}}, \quad (5.8)$$

where $\alpha \in (0, 1]$ is a step length. Based on (5.5) and (5.7), we are able to approximate the gradient of the variational lower bound with respect to the posterior mean, and the Hessian of the variational objective as

$$\frac{\partial \mathcal{F}}{\partial \mathbf{m}} \approx \frac{1}{\sigma_g^2} \mathbf{A}(\mathbf{y} - \mathbf{A}\mathbf{m} - \mathbf{b}) - \mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1} \mathbf{m}, \quad \frac{\partial^2 \mathcal{F}}{\partial \mathbf{m} \partial \mathbf{m}^\top} \approx -\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1} - \frac{1}{\sigma_g^2} \mathbf{A}\mathbf{A}.$$

Once (5.8) has converged to its optimum, declared as \mathbf{m}^+ , we are able to calculate the approximate posterior covariance matrix,

$$\mathbf{C} = \left(\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1} + \frac{1}{\sigma_g^2} \mathbf{A}\mathbf{A} \right)^{-1}. \quad (5.9)$$

Note that the entries of matrix \mathbf{A} depend on the optimum mean \mathbf{m}^+ .

Hyperparameter learning

Using the optimum posterior parameters found in the above section, the approximated lower bound reduces to

$$\mathcal{F}_\theta \approx -\frac{1}{2} N \log 2\pi\sigma_g^2 - \frac{\mathbf{e}^\top \mathbf{e}}{2\sigma_g^2} - \frac{1}{2} \left[\mathbf{m}^\top \mathbf{C}^{-1} \mathbf{m} - \log |\mathbf{C}| + \log |\mathbf{K}_{\mathbf{f},\mathbf{f}}| \right]. \quad (5.10)$$

Unfortunately, because the posterior moments depend of the hyperparameters (covariance function hyperparameters and noise variances), we are unable to use optimisation techniques based on partial derivatives with the aim of finding the optimal set of hyperparameters. Thus, as in [Steinberg and Bonilla \(2014\)](#), we resort to a derivative-free optimisation method known as BOBYQUA (Bound Optimization BY Quadratic Approximation) and introduced in [Powell \(2009\)](#). Additionally, note that for each iteration of the optimisation procedure, we require to perform, as an inner loop, the Newton's method in order to find the optimal posterior moments, as described in Algorithm 4.

Algorithm 4 Evaluation of the objective function for the Wiener LFM.

- 1: **Input:** Training data: \mathbf{t} and \mathbf{y} . Hyper-parameters $\boldsymbol{\theta}$, and variational factors \mathbf{m} and \mathbf{C} .
 - 2: **repeat**
 - 3: Update \mathbf{m} using Newton’s method (5.8).
 - 4: **until** (5.8) has converged.
 - 5: Update \mathbf{C} according to (5.9).
 - 6: **Return:** Evaluation of the lower bound (5.10)
-

5.2.2 Predictive distribution

The predictive distribution for the response function, \mathbf{f}^* , at unknown time stamps \mathbf{t}^* , can be obtained by evaluating

$$p(\mathbf{f}^*|\mathbf{y}) = \int p(\mathbf{f}^*|\mathbf{f})q(\mathbf{f}) \, d\mathbf{f},$$

which can be straightforwardly calculated as

$$p(\mathbf{f}^*|\mathbf{y}) = \mathcal{N}\left(\mathbf{K}_{\mathbf{f}^*,\mathbf{f}}\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1}\mathbf{m}, \mathbf{K}_{\mathbf{f}^*,\mathbf{f}^*} - \mathbf{K}_{\mathbf{f}^*,\mathbf{f}}\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1}[\mathbf{I}_N - \mathbf{C}\mathbf{K}_{\mathbf{f},\mathbf{f}}]\mathbf{K}_{\mathbf{f}^*,\mathbf{f}}^\top\right).$$

Additionally, we can find the predicted observations,

$$p(\mathbf{y}^*|\mathbf{y}) = \int g(\mathbf{f}^*)p(\mathbf{f}^*|\mathbf{y}) \, d\mathbf{f}^*,$$

using quadrature. Interestingly, we are also able to estimate the predictive distribution for the latent force function, \mathbf{u}^* , using

$$p(\mathbf{u}^*|\mathbf{y}) = \int p(\mathbf{u}^*|\mathbf{f})q(\mathbf{f}|\mathbf{m}, \mathbf{C}) \, d\mathbf{f},$$

which becomes

$$p(\mathbf{u}^*|\mathbf{y}) = \mathcal{N}\left(\mathbf{K}_{\mathbf{u}^*,\mathbf{f}}\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1}\mathbf{m}, \mathbf{K}_{\mathbf{u}^*,\mathbf{u}^*} - \mathbf{K}_{\mathbf{u}^*,\mathbf{f}}\mathbf{K}_{\mathbf{f},\mathbf{f}}^{-1}[\mathbf{I}_N - \mathbf{C}\mathbf{K}_{\mathbf{f},\mathbf{f}}]\mathbf{K}_{\mathbf{u}^*,\mathbf{f}}^\top\right).$$

5.2.3 Related work

The first non-linear dynamic system approximated using the LFM framework was introduced in Lawrence et al. (2006). Specifically, gene expression data was approximated using a Hammerstein system. Nevertheless, the convolution operation was approximated by sums in order to keep the inference process tractable. In contrast, we are able to build

all types of non-linear dynamic systems using SLFMs. As demonstrated in [Hartikainen et al. \(2012\)](#), Wiener, Hammerstein and Drift non-linear systems can be approximated using the SLFM approach.

However, we remark that the non-linear LFM, described in 5.3, can be also approached using Warped GPs ([Lázaro-Gredilla, 2012](#); [Snelson et al., 2004](#)) or Black-box likelihoods ([Dezfouli and Bonilla, 2015](#)).

5.2.4 Experiments

In this section we explore the ability of the proposed approach to recover the latent force function (excitation function) from the noisy observations of the transformed response of a LTI system. Unfortunately, we are unable to guarantee that the latent force is accurately estimated. As argued in [Davies and Husmeier \(2014\)](#), given that the latent force is unobservable, we are only able to estimate the shape of the latent force function. Furthermore, recall that the proposed method induces uncertainty from the observed outputs to the response function (because of the linearisation of the static non-linear function). Since we adopt two different linearisation methods, we refer to the Wiener system approximated using the Statistical and Taylor series linearisation methods as LFM-S and LFM-T, respectively.

For the experiments, we use a single-input single-output second order LFM, as described in section 2.2.3, with parameters $B = 1$, $C = 3$, $S = 5$ and $l = 0.5$. From this LFM, we sample 400 data points of the response and excitation functions in the interval $[0, 3]$ s. We build the observation vector \mathbf{y} by transforming the response data \mathbf{f} using the static non-linear function $g(f)$, and then adding a white noise process with variance 0.01. Next, we divide the dataset into 153 data points for training and 247 data points for testing. The proposed approach is learned using 10 different initialisations, and the one that achieved the highest lower bound value is selected as the final solution.

We consider four different static non-linear functions, which are listed in the first column of Table 5.1. In this table we compare the performance of both linearisation methods, regarding the estimation of the response function $f(t)$ at the testing points, using the measurements NMSE and NLPD described in appendix A. Since the first non-linear static function has no gradients w.r.t. f , we are unable to use the Taylor series approach to approximate the Wiener system. In consequence, there are no performance measurements reported for that case. According to the results listed in Table 5.1, we notice that the statistical linearisation method outperformed the Taylor series approach, except for the case when $g(f) = \sin(f)$. Furthermore, these results indicate that the

Table 5.1: Comparison of the Statistical and Taylor series linearisation methods based on NMSE and NLPD performance measurements for the prediction of the response function $f(t)$ at testing data.

$g(f)$	LFM-S		LFM-T	
	NMSE	NLPD	NMSE	NLPD
$2\text{sign}(f) + f^3$	3.8×10^{-4}	-4.3618	-	-
$f^3 + f^2 + f$	6.8×10^{-6}	-5.6307	9.9×10^{-3}	-1.8531
$\exp(f)$	2.8×10^{-5}	-4.7888	3.5×10^{-5}	-4.6239
$\sin(f)$	3.8×10^{-3}	1.4255	1.4×10^{-4}	-4.2264

amplitude values are not only well fitted (small values of NMSE), and also its variability is adequately described by the predictive distribution (negative values of NLPD).

Figure 5.2 shows the forcing functions estimated for each Wiener system considered in Table 5.1. Notice that the forcing functions obtained from the worst fitted response functions (i.e. LFM-T at the second case, and LFM-S at the fourth considered Wiener system), are smoother than the true function. In contrast, we can see a coarser behaviour (but matched amplitudes) of the estimated forcing functions in the first and third Wiener systems using the LFM-S and LFM-T approaches, respectively. Although the rest of the forcing functions have different amplitudes (w.r.t. the true function), their waveform matches (correlates) the shape of the true forcing function.

5.3 Wiener system estimation based on sequential Laguerre processes

In this proposed approach, we are interested on estimating the impulse response function of Wiener systems by using the Laguerre functions. We start by defining the continuous model with its linear and non-linear parts. Then, the linear dynamic part of a Wiener system can be described by the following continuous state-space model

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\epsilon(t), \quad [f(t), u(t)]^\top = \mathbf{C}\mathbf{x}(t), \quad (5.11)$$

where \mathbf{A} , \mathbf{B} , \mathbf{C} and $\mathbf{w}(t)$ follow the same forms defined in section 4.3 (for $Q = 1$ and $D = 1$). The static non-linear function is included at the following observation model,

$$\mathbf{y}(t) = g(\mathbf{C}\mathbf{x}(t)) + \mathbf{r}(t), \quad (5.12)$$

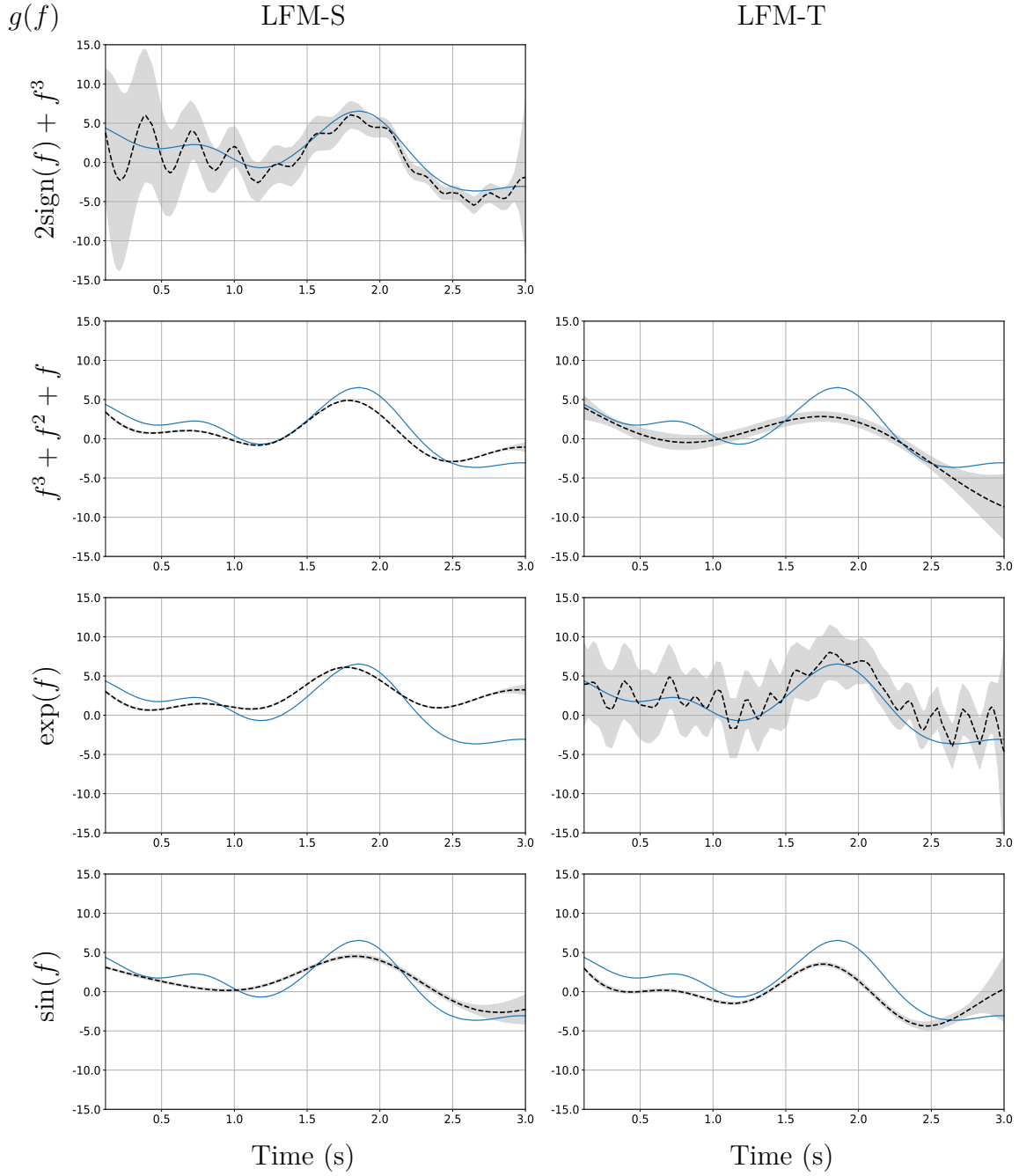


Figure 5.2: Plot of the forcing functions $u(t)$, true function (blue line), and the mean (black dashed line) and two times the standard deviation (grey shade) of LFM's predictions for each Wiener system considered in experiments 5.2.4.

with $\mathbf{r}(t)$ modelling the noise processes for the input and output functions. Note that the evaluation of the non-linear function given in (5.12) is only applied over $f(t)$, i.e. $g(\mathbf{C}\mathbf{x}(t)) = [g(f(t)), u(t)]^\top$.

5.3.1 Inference

In order to perform the inference procedure, we require to transform the continuous model given in (5.11) and (5.12) into its discrete time analogous. Fortunately, the linear part, given in (5.11), can be fully described by the discrete model developed in (4.7). In contrast, the discrete observation model defined as

$$\mathbf{y}_k = g(\mathbf{C}\mathbf{x}_k) + \mathbf{r}_k, \quad (5.13)$$

includes the non-linear static function. Hence, in order to have a tractable inference procedure for the model described above, we adopt the extended and the unscented Kalman filter approaches (Särkkä, 2013).

Extended Kalman filter (EKF)

Here, we approximate the non-linear function by using the Taylor series linearisation described in section 5.1. In consequence, the *update steps* of the standard Kalman filter, described in algorithm 1, are changed by the following expressions

$$\begin{aligned} \mathbf{v}_k &= \mathbf{y}_k - g(\mathbf{C}\mathbf{m}_k^-), \\ \mathbf{S}_k &= \mathbf{J}_k \mathbf{P}_k^- \mathbf{J}_k^\top + \mathbf{R}_k, \\ \mathbf{K}_k &= \mathbf{P}_k^- \mathbf{J}_k^\top \mathbf{S}_k^{-1}, \\ \mathbf{m}_k &= \mathbf{m}_k^- + \mathbf{K}_k \mathbf{v}_k, \\ \mathbf{P}_k &= \mathbf{P}_k^- - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^\top, \end{aligned}$$

where the Jacobian matrix is defined as

$$\mathbf{J}_k = \left. \frac{\partial g(\mathbf{C}\mathbf{x}_k)}{\partial \mathbf{x}_k} \right|_{\mathbf{x}_k = \mathbf{m}_k^-}.$$

Unscented Kalman filter (UKF)

In this case, we approximate the non-linear function by using the Statistical linearisation method described in section 5.1. In consequence, the *update steps* of the standard

Kalman filter, described in algorithm 1, are changed by the following expressions

$$\begin{aligned}\mathbf{K}_k &= \mathbf{C}_k \mathbf{S}_k^{-1}, \\ \mathbf{m}_k &= \mathbf{m}_k^- + \mathbf{K}_k [\mathbf{y}_k - \boldsymbol{\mu}_k], \\ \mathbf{P}_k &= \mathbf{P}_k^- - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^\top,\end{aligned}$$

where $\boldsymbol{\mu}_k$ and \mathbf{C}_k are the predictive mean and cross-covariance matrix calculated using the expression given in equation (5.2). The predicted covariance matrix of the state is calculated as

$$\mathbf{S}_k = \sum_{i=0}^{2D} c_i (\mathcal{Y}_i - \boldsymbol{\mu}_k) (\mathcal{Y}_i - \boldsymbol{\mu}_k)^\top.$$

Hyperparameter learning

The model parameters, for both Extended and Unscented Kalman filters, are learned by maximizing the logarithm of the likelihood function, which is given by

$$\begin{aligned}\underset{\boldsymbol{\theta}_{\text{SLP}}}{\text{maximize}} \quad \log p(\mathbf{y}|\mathbf{t}) &= -\frac{1}{2} \sum_{k=1}^K \left[(\mathbf{y}_k - g(\mathbf{C}\mathbf{m}_k))^\top \mathbf{S}_k^{-1} (\mathbf{y}_k - g(\mathbf{C}\mathbf{m}_k)) \right. \\ &\quad \left. + \log |2\pi \mathbf{S}_k| \right],\end{aligned}\tag{5.14}$$

where $\boldsymbol{\theta}_{\text{SLP}}$ comprises the hyperparameters required to describe the model defined in 5.11 and 5.12. Besides, note that \mathbf{S}_k and \mathbf{m}_k depend on $\boldsymbol{\theta}_{\text{SLP}}$.

5.3.2 Experiments

In this section, we are interested in analysing the ability of the proposed approach to estimate the impulse response function of Wiener systems by means of the Laguerre functions. First, we generated 200 data points equally spaced along the range $[0, 8]$ s, by sampling $u(t)$ from a GP based on a square exponential covariance function with lengthscale equal to one. Then, the response data $\{f_k\}_{k=1}^{200}$ is obtained by applying $\{u_k\}_{k=1}^{200}$ through a second order dynamical system characterized by the following IRF

$$G(t) = \frac{1}{\omega} \exp\left(-\frac{b_1 t}{2}\right) \sinh(\omega t),$$

with $b_0 = 2$, $b_1 = 3$ and $\omega = \sqrt{b_1^2 - 4b_0}/2$. Then, the observed output data $\{\mathbf{y}_k\}_{k=1}^{200}$ is generated by adding white noise (with variance $\sigma^2 = 0.01$) to the transformed response values, as described in (5.13). The non-linear static functions considered in the experiments are listed in the first column of Table 5.2.

The proposed approach is learned from 10 different initializations using the procedure described in 5.3.1, and all the available data (i.e. all samples are used for training the model). In Figure 5.3 and Table 5.2 are summarized the overall performance of the proposed model to estimate the IRF regarding each non-linear static function. The standard deviation values, listed in Table 5.2, indicate that the solutions found by the EKF for a specific non-linear function highly differ one from another. Nevertheless, the mean values for the UKF show that its main trend, from the estimated IRFs, adequately fitted the true IRF.

Table 5.2: Mean and standard deviation of the NMSE values obtained from the 10 IRFs learned for experiment 5.3.2.

$g(f)$	EKF		UKF	
	mean	std	mean	std
$f^3 + f^2 + f$	0.0763	0.2388	0.0035	0.0099
$\exp(f)$	0.0017	0.0043	0.0016	0.0032
$\sin(f)$	0.1809	0.3049	0.1044	0.2055

For example, most of the IRFs, estimated at the exponential transformation case, highly matched the form of the true IRF.

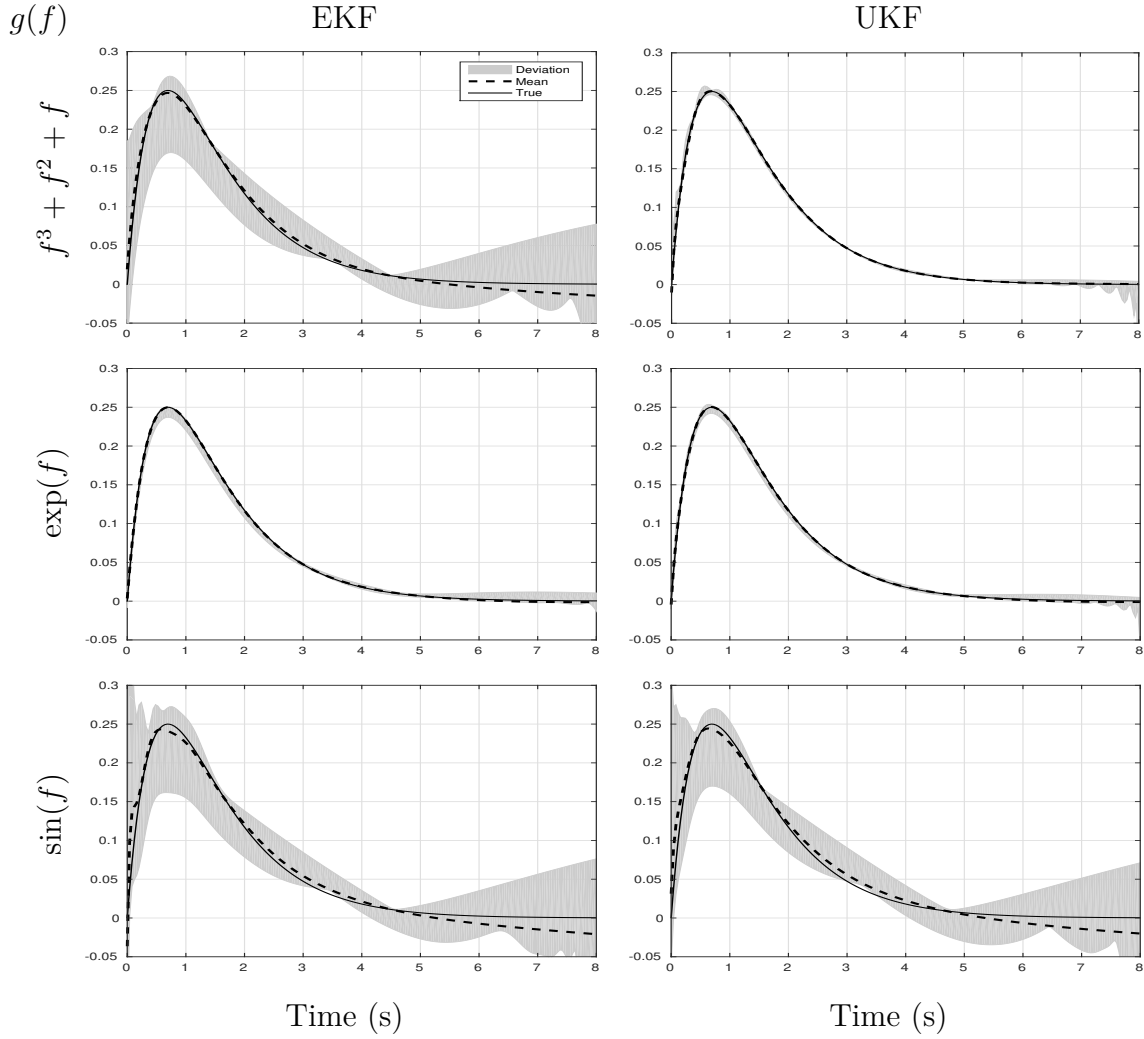


Figure 5.3: Comparison between the true IRF, and the mean and 2 times standard deviation of IRFs estimated by the proposed approach.

5.4 Conclusions

In this chapter, we have proposed two different approaches based on Wiener systems. The first approach is aimed to estimate the latent force function using LFMs. From the experiments, it is evidenced that the proposed approach is able to estimate the shape of the latent force from the noise corrupted observations of Wiener system outputs.

On the other hand, the second proposed approach estimates the IRF using Laguerre functions and SLFMs. The experiments demonstrated that the proposed model is able to accurately estimate the IRFs of Wiener systems in the presence of noise. Furthermore,

we could consider using the statistical linearisation in order to deal with non-linear functions from which gradients are intractable.

Chapter 6

Conclusions and Future Work

This chapter summarizes the contributions and research work done in the thesis, besides some future research lines are presented.

6.1 Conclusions

This thesis focused on developing new approaches and extensions for Latent Force models.

Number of latent forces. In Chapter 3 a variational framework was developed, that allows to approximate the posterior of the binary matrix modelled by an Indian Buffet process for multiple-output LFMs. From this posterior we are not only able to estimate the number of latent functions required to explain the observed data, but also to estimate the sparse structure relating the input and output functions. This variational approach was successfully applied on gene expression, motion capture and weather datasets. Specifically, for the gene expression data we were able to estimate the correct number of transcription factors.

Modelling multiple-input multiple-output data. Chapter 4 extended LFMs and SLFMs usage. Instead of focusing on the estimation of the latent forces, the proposed approaches (CLPs and SLPs) were aimed to model multiple-input multiple-output data. By capturing the correlations among the inputs and outputs we are able to point-estimate the impulse response functions (IRFs) using Laguerre functions. These approaches were compared regarding their performance on the estimation of the IRF and predictions of missing data in multiple-input multiple-output scenarios. From the experiments, we

evidenced that the CLP approach performed better than the SLP, due to the former uses the correlation from the whole training data in order to make predictions at any time.

Approximation of Wiener systems. In Chapter 5 two different approaches were developed. In the first approach, we considered the estimation of the latent forces on Wiener systems. The problem imposed by the static non-linear function is addressed using linearisation techniques. The approximated linear model is evaluated at the mean of the posterior of the response function. Besides, the posterior of the response function is approached using variational inference. We demonstrated the capacity of the proposed approach to estimate the form of the latent forces.

On the other hand, the second approach was instead aimed to estimate the impulse response function of a Wiener system. In this approach, the extended Kalman filter was applied over sequential latent force models in order to deal with non-linear static functions. We demonstrated that this approach successfully estimated the impulse response function for different non-linear static functions.

6.2 Future Work

Here we discuss some potential research lines.

Number of latent functions. The main drawback for the variational approach proposed in Chapter 3 is the requirement of the truncation level in order to have a tractable model. Nevertheless, we are able to use Markov chain Monte Carlo (MCMC) steps aimed to find the number of latent functions, as in [Chatzis and Kosmopoulos \(2015\)](#); [Knowles and Ghahramani \(2011\)](#); [Teh \(2007\)](#). Hence, we could resort to a Hybrid inference, where the number of latent functions is estimated using the MCMC step and the rest of the random variables are estimated using the proposed variational approach. However, we must take special care with the MCMC step because for each latent force that is added, we require to select the length-scale and the sensitivity values.

Modelling multiple-input multiple-output data. As discussed in Chapter 4, the main issue of the CLP approach is that we approximate the convolutions (and hence the covariance functions) using discrete sums. This problem can be addressed by using a set of orthonormal functions from which the convolutions have closed forms or approximat-

ing the squared exponential covariance functions using Kernel Fourier features ([Rahimi and Recht, 2008](#)).

Besides, we could consider to add more flexibility to the proposed approaches, by assuming that each input function contributes to each output using a different linear system. Figure 6.1 depicts how each input function generates a different response from different IRFs, which are added to obtain the output function.

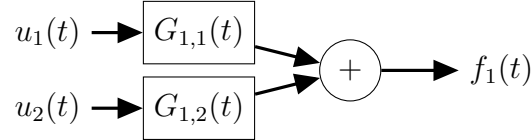


Figure 6.1: Block representation of a mutiple-input single-output system.

Non-linear dynamical systems. From the approach presented in section 5.3, we are also able to model another non-linear dynamical systems, such as, Hammerstein systems and the combination Wiener-Hammerstein dynamical system. This can be done by extending the examples presented in [Hartikainen et al. \(2012\)](#) using the SLP approach.

On the other hand, we are able to combine the variational approach proposed to estimate the number of latent functions, with the approach proposed in section 5.2. Hence, this new model would be able to infer the number and form of latent forces for Wiener systems. Additionally, we could consider to model multiple output data, with the aim of increasing the accuracy on estimating the latent force (excitation) function. This can be done by considering the work presented in [Bonilla et al. \(2016\)](#).

Appendix A

Performance metrics

In order to compare the results obtained by using different configurations or methods, we evaluate the performance of predicting missing data using the normalised mean square error (NMSE) and the negative log probability density (NLPD) (Tan et al., 2016). These measurements are defined per output as

$$\text{NMSE} = \frac{\frac{1}{N_d^*} \sum_{j=1}^{N_d^*} (y_{d,j}^* - \mu_{d,j}^*)^2}{\frac{1}{N_d^*} \sum_{j=1}^{N_d^*} (y_{d,j}^* - \bar{y}_d)^2},$$

$$\text{NLPD} = \frac{1}{2N_d^*} \sum_{j=1}^{N_d^*} \left[\frac{(y_{d,j}^* - \mu_{d,j}^*)^2}{\sigma_{d,j}^{*2}} + \log(2\pi\sigma_{d,j}^{*2}) \right],$$

where $y_{d,j}^*$ is the j -th true test value of output d . Similarly, $\mu_{d,j}^*$ and $\sigma_{d,j}^{*2}$ are the mean and variance values of the predictive distribution for the d -th output at the j -th test time, respectively. Meanwhile, \bar{y}_d is the average value of the training values \mathbf{y}_d .

Appendix B

Extension for the estimation of the number of latent forces

B.1 Lower Bound terms description

Lower bound term defined in section 3.3 are mathematically described next:

$$\begin{aligned}\mathbb{E}[p(\mathbf{y}|-)] &= \sum_{q=1}^{Q_+} \text{tr}(\mathbf{m}_q \mathbb{E}[\mathbf{u}_q^\top]) - \frac{1}{2} \sum_{q=1}^{Q_+} \sum_{q'=1}^{Q_+} \text{tr}(\mathbf{P}_{q,q'} \mathbb{E}[\mathbf{u}_{q'} \mathbf{u}_q^\top]) \\ &\quad - \frac{1}{2} \sum_{q=1}^{Q_+} \text{tr}(\mathbf{K}_{\mathbf{u}_q, \mathbf{u}_q}^{-1} \mathbb{E}[\mathbf{u}_q \mathbf{u}_q^\top]) - \frac{1}{2} \sum_{q=1}^{Q_+} \log |\mathbf{K}_{\mathbf{u}_q, \mathbf{u}_q}| \\ &\quad + \frac{1}{2} \sum_{d=1}^D N_d \log \beta_d - \frac{1}{2} \sum_{d=1}^D \beta_d \mathbf{y}_d^\top \mathbf{y}_d - \frac{1}{2} \sum_{d=1}^D \sum_{q=1}^{Q_+} \eta_{d,q} c_{d,q} \\ &\quad - \frac{MQ_+}{2} \log 2\pi - \frac{N}{2} \log 2\pi,\end{aligned}$$

$$\begin{aligned}\mathbb{E}[\log p(\mathbf{Z}|\mathbf{v})] &= -\frac{1}{2} \log 2\pi \sum_{d=1}^D \sum_{q=1}^{Q_+} \mathbb{E}[Z_{d,q}] + \sum_{d=1}^D \sum_{q=1}^{Q_+} \mathbb{E}[Z_{d,q}] \mathbb{E}[\log \pi_q] \\ &\quad + \sum_{d=1}^D \sum_{q=1}^{Q_+} (1 - \mathbb{E}[Z_{d,q}]) \mathbb{E}[\log(1 - \pi_q)],\end{aligned}$$

$$\mathbb{E}[\log p(\mathbf{v})] = (\alpha - 1) \sum_{q=1}^{Q_+} [\psi(\tau_{q1}) - \psi(\tau_{q1} + \tau_{q2})] + \log(\alpha) Q_+,$$

with

$$\mathbb{E}[\mathbf{u}_{q'}\mathbf{u}_q^\top] = \widetilde{\mathbf{K}}_{\mathbf{u}_{q'},\mathbf{u}_q} + \widetilde{\mathbf{u}}_{q'}\widetilde{\mathbf{u}}_q^\top,$$

Meanwhile, the entropies are defined as

$$\begin{aligned} H(\mathbf{v}) &= \sum_{q=1}^{Q_+} \left[\log \left(\frac{\Gamma(\tau_{q1})\Gamma(\tau_{q2})}{\Gamma(\tau_{q1} + \tau_{q2})} \right) - (\tau_{q1} - 1)\psi(\tau_{q1}) \right. \\ &\quad \left. - (\tau_{q2} - 1)\psi(\tau_{q2}) + (\tau_{q1} + \tau_{q2} - 2)\psi(\tau_{q1} + \tau_{q2}) \right], \\ H(\mathbf{Z}) &= - \sum_{d=1}^D \sum_{q=1}^{Q_+} [(1 - \eta_{d,q}) \ln(1 - \eta_{d,q}) + \eta_{d,q} \ln \eta_{d,q}], \end{aligned}$$

and

$$H(\mathbf{u}) = \frac{Q_+M}{2}(1 + \log(2\pi)) + \frac{1}{2} \sum_{q=1}^Q \ln |\widetilde{\mathbf{K}}_{\mathbf{u}_q,\mathbf{u}_q}|,$$

where $\Gamma(\cdot)$ is the gamma function.

B.2 Predictive distribution for latent forces

Let us assume we are interested in predicting the values $\mathbf{u}^* = [\mathbf{u}_1^{*\top}, \dots, \mathbf{u}_{Q_+}^{*\top}]^\top$ at testing inputs $\mathbf{t}^* = [\mathbf{t}_1^{*\top}, \dots, \mathbf{t}_{Q_+}^{*\top}]^\top$. Thus, we are able to approximate the predictive distribution for \mathbf{u}^* as

$$p(\mathbf{u}^*|\mathbf{y}) = \int p(\mathbf{u}^*|\mathbf{u})q(\mathbf{u})d\mathbf{u}.$$

where $p(\mathbf{u}^*|\mathbf{u}) = \mathcal{N}(\mathbf{u}^*|\mathbf{K}_{\mathbf{u}^*,\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{u}, \mathbf{K}_{\mathbf{u}^*,\mathbf{u}^*} - \mathbf{K}_{\mathbf{u}^*,\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{u},\mathbf{u}^*})$ and $q(\mathbf{u})$ is defined in section 3.3.3. Note that the above integral is tractable since both probabilities are normally distributed. Thus, the above predictive distribution is reduced to

$$p(\mathbf{u}^*|\mathbf{y}) = \mathcal{N}(\mathbf{u}^*|\boldsymbol{\mu}_{\mathbf{u}^*}, \boldsymbol{\Sigma}_{\mathbf{u}^*}),$$

with

$$\begin{aligned} \boldsymbol{\mu}_{\mathbf{u}^*} &= \mathbf{K}_{\mathbf{u}^*,\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\widetilde{\mathbf{u}}, \\ \boldsymbol{\Sigma}_{\mathbf{u}^*} &= \mathbf{K}_{\mathbf{u}^*,\mathbf{u}^*} - \mathbf{K}_{\mathbf{u}^*,\mathbf{u}} \left(\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1} - \mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\widetilde{\mathbf{K}}_{\mathbf{u},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1} \right) \mathbf{K}_{\mathbf{u},\mathbf{u}^*}. \end{aligned}$$

References

- Mauricio Álvarez and Neil D. Lawrence. Sparse Convolved Gaussian Processes for Multi-output Regression. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 57–64. Curran Associates, Inc., 2009. URL <http://papers.nips.cc/paper/3553-sparse-convolved-gaussian-processes-for-multi-output-regression.pdf>. (page 41)
- Mauricio A. Álvarez, David Luengo, and Neil D. Lawrence. Latent Force Models. In David van Dyk and Max Welling, editors, *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, pages 9–16, Clearwater Beach, Florida, 16-18 April 2009. JMLR W&CP 5. (page 15)
- Mauricio A. Álvarez, David Luengo, Michalis K. Titsias, and Neil D. Lawrence. Variational Inducing Kernels for Sparse Convolved Multiple Output Gaussian Processes. Technical report, University of Manchester, 2009. (pages 3, 21, 25, and 37)
- Mauricio A. Álvarez, Lorenzo Rosasco, and Neil D. Lawrence. Kernels for Vector-Valued Functions: a Review. *Foundations and Trends® in Machine Learning*, 4(3):195–266, 2012. (page 2)
- Mauricio A. Álvarez, David Luengo, and Neil D. Lawrence. Linear Latent Force Models Using Gaussian Processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2693–2705, 2013. (pages v, 1, 11, and 14)
- Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. (pages 26, 27, 28, and 60)
- Edwin Bonilla, Daniel Steinberg, and Alistair Reid. Extended and Unscented Kitchen Sinks. In *International Conference on Machine Learning*, New York, jun 2016. (pages 62 and 74)
- Phillip Boyle and Marcus Frean. Dependent Gaussian Processes. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 217–224. MIT Press, 2005. URL <http://papers.nips.cc/paper/2561-dependent-gaussian-processes.pdf>. (page 11)
- Kian M. Chai, Christopher Williams, Stefan Klanke, and Sethu Vijayakumar. Multi-task Gaussian Process Learning of Robot Inverse Dynamics. In *NIPS 2008*, <http://eprints.pascal-network.org/archive/00004640/>, 2009. (page 32)

- S. P. Chatzis and D. Kosmopoulos. A Nonparametric Bayesian Approach toward Stacked Convolutional Independent Component Analysis. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2803–2811, Dec 2015. doi: 10.1109/ICCV.2015.321. (pages 44 and 73)
- Zhenwen Dai, Andreas C. Damianou, James Hensman, and Neil D. Lawrence. Gaussian Process Models with Parallelization and GPU acceleration. *CoRR*, abs/1410.4984, 2014. URL <http://arxiv.org/abs/1410.4984>. (pages 21 and 27)
- Vinny Davies and Dirk Husmeier. Modelling transcriptional regulation with Gaussian processes. In Andre X.C.N Valente, Abhijit Sarkar, and Yuan Gao, editors, *Recent Advances in Systems Biology Research*, volume 1, pages 157–184. Nova Science Publishers, 2014. ISBN 978-1-629-48736-6. (page 64)
- B. De Moor, P. De Gersem, B. De Schutter, and W. Favoreel. DAISY: A database for identification of systems. *Department of Electrical Engineering, ESAT/STADIUS, KU Leuven, Belgium. CD-player arm, Mechanical systems, 96-007*, 1997. URL <http://homes.esat.kuleuven.be/~smc/daisy/>. (page 55)
- Amir Dezfouli and Edwin V Bonilla. Scalable Inference for Gaussian Process Models with Black-Box Likelihoods. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1414–1422. Curran Associates, Inc., 2015. (pages 59 and 64)
- Finale Doshi-Velez, Kurt Miller, Jurgen Van Gael, and Yee Whye Teh. Variational inference for the Indian Buffet process. In *AISTATS 2009*, pages 137–144, 2009. (pages 3, 25, 26, and 28)
- Dean G Duffy. *Green’s functions with applications*. CRC Press, Abingdon, 2015. (pages 7, 13, and 15)
- Yarin Gal, Mark van der Wilk, and Carl Rasmussen. Distributed Variational Inference in Sparse Gaussian Process Regression and Latent Variable Models. In Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3257–3265. Curran Associates, Inc., 2014. (pages 21 and 27)
- Thore Graepel. Solving Noisy Linear Operator Equations by Gaussian Processes: Application to Ordinary and Partial Differential Equations. In *Proceedings of the Twentieth International Conference on Machine Learning*, January 2003. (page 11)
- Thomas L. Griffiths and Zoubin Ghahramani. Infinite Latent Feature Models and the Indian Buffet Process. In *Advances in Neural Information Processing Systems*, pages 475–482. MIT Press, 2005. (pages 3 and 24)
- Thomas L. Griffiths and Zoubin Ghahramani. The Indian Buffet Process: An Introduction and Review. *Journal of Machine Learning Research*, 12:1185–1224, July 2011. ISSN 1532-4435. URL <http://dl.acm.org/citation.cfm?id=1953048.2021039>. (page 3)

- Cristian Guarnizo and Mauricio A. Álvarez. Impulse Response Estimation of Linear Time-Invariant systems using Convolved Gaussian Processes and Laguerre functions. In Marcelo Mendoza and Sergio Velastín, editors, *To appear in Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications: 22nd Iberoamerican Congress, CIARP 2017, Valparaiso, Chile, November 7-10, 2017, Proceedings*, pages 635–642, Cham, 2017. Springer International Publishing. ISBN 978-3-319-25751-8. (page 5)
- Cristian Guarnizo, Mauricio A. Álvarez, and Alvaro A. Orozco. Indian Buffet Process for Model Selection in Latent Force Models. In Alvaro Pardo and Josef Kittler, editors, *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications: 20th Iberoamerican Congress, CIARP 2015, Montevideo, Uruguay, November 9-12, 2015, Proceedings*, pages 635–642, Cham, 2015. Springer International Publishing. ISBN 978-3-319-25751-8. (pages 5, 32, and 33)
- Robert Haber and László Keviczky. *Nonlinear System Identification - Input-Output Modeling Approach*. Kluwer Academic, 1999. (page 46)
- Jouni Hartikainen and Simo Särkkä. Kalman Filtering and Smoothing Solutions to Temporal Gaussian Process Regression Models. In *IEEE International Workshop on Machine Learning for Signal Processing*, pages 379–384, August 2010. (pages 9, 17, 18, and 49)
- Jouni Hartikainen and Simo Särkkä. Sequential Inference for Latent Force Models. In Fábio Gagliardi Cozman and Avi Pfeffer, editors, *UAI 2011, Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence, Barcelona, Spain, July 14-17, 2011*, pages 311–318. AUAI Press, 2011. ISBN 978-0-9749039-7-2. URL https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=2238&proceeding_id=27. (pages 4, 17, and 18)
- Jouni Hartikainen, Mari Seppänen, and Simo Särkkä. State-Space Inference for Non-Linear Latent Force Models with Application to Satellite Orbit Prediction. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*. icml.cc / Omnipress, 2012. URL <http://icml.cc/2012/papers/477.pdf>. (pages 17, 22, 64, and 74)
- Brett W. Israelsen and Dale A. Smith. Generalized Laguerre Reduction of the Volterra kernel for Practical Identification of Nonlinear Dynamic Systems. *CoRR*, abs/1410.0741, 2014. URL <http://arxiv.org/abs/1410.0741>. (pages 45 and 46)
- David A. Knowles and Zoubin Ghahramani. Nonparametric Bayesian Sparse Factor Models with application to Gene Expression modelling. *Annals of Applied Statistics*, 5(2B):1534–1552, 2011. (pages 43, 44, and 73)
- Neil D. Lawrence, Guido Sanguinetti, and Magnus Rattray. Modelling transcriptional regulation using Gaussian Processes. In *Neural Information Processing Systems*, pages 785–792, 2006. (pages 11, 14, 22, 36, 47, and 63)
- Miguel Lázaro-Gredilla. Bayesian Warped Gaussian Processes. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1619–1627. Curran Associates, Inc., 2012. (page 64)

- Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012. ISBN 0262018020, 9780262018029. (page 43)
- Trung V. Nguyen and Edwin V. Bonilla. Collaborative Multi-output Gaussian Processes. In Nevin L. Zhang and Jin Tian, editors, *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence, UAI 2014, Quebec City, Quebec, Canada, July 23-27, 2014*, pages 643–652. AUAI Press, 2014. ISBN 978-0-9749039-1-0. URL https://dslpitt.org/uai/displayArticles.jsp?mmnu=1&smnu=1&proceeding_id=30. (page 41)
- Manfred Oppner and Guido Sanguinetti. Learning combinatorial transcriptional dynamics from gene expression data. *Bioinformatics*, 26(13):1623, 2010. doi: 10.1093/bioinformatics/btq244. URL [+http://dx.doi.org/10.1093/bioinformatics/btq244](http://dx.doi.org/10.1093/bioinformatics/btq244). (pages 36 and 38)
- Michael A. Osborne, Stephen J. Roberts, Alex Rogers, Sarvapali D. Ramchurn, and Nicholas R. Jennings. Towards Real-Time Information Processing of Sensor Network Data Using Computationally Efficient Multi-output Gaussian Processes. In *IPSN*, pages 109–120. IEEE Computer Society, 2008. ISBN 978-0-7695-3157-1. URL <http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=4505448>. (page 41)
- M. J. D. Powell. The BOBYQA algorithm for bound constrained optimization without derivatives. Technical report, Department of Applied Mathematics and Theoretical Physics, Cambridge, England, 2009. (page 62)
- Ali Rahimi and Benjamin Recht. Random Features for Large-Scale Kernel Machines. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1177–1184. Curran Associates, Inc., 2008. URL <http://papers.nips.cc/paper/3182-random-features-for-large-scale-kernel-machines.pdf>. (page 74)
- Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Machine learning of linear differential equations using Gaussian processes. *Journal of Computational Physics*, 348:683 – 693, 2017. ISSN 0021-9991. doi: <http://dx.doi.org/10.1016/j.jcp.2017.07.050>. URL <http://www.sciencedirect.com/science/article/pii/S0021999117305582>. (page 11)
- Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006. ISBN 0-262-18253-X. (pages 2, 8, 9, 48, and 51)
- B. C. Reginato and G. H. C. Oliveira. On selecting the MIMO Generalized Orthonormal Basis Functions poles by using Particle Swarm Optimization. In *Control Conference (ECC), 2007 European*, pages 5182–5188, July 2007. (page 45)
- Ricardo S. Risuelo, Giulio Bottegal, and Hakan Hjalmarsson. Kernel-based system identification from noisy and incomplete input-output data. In *arXiv:1605.03733*, May 2016. (page 51)
- Simo Särkkä. *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013. ISBN 1107619289, 9781107619289. (pages 4, 19, 20, and 67)

- Edward Snelson, Zoubin Ghahramani, and Carl E. Rasmussen. Warped Gaussian Processes. In S. Thrun, L. K. Saul, and P. B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, pages 337–344. MIT Press, 2004. (page 64)
- Rafal Stanislawski, Wojciech P. Hunek, and Krzysztof J. Latawiec. Modeling of non-linear block-oriented systems using orthonormal basis and radial basis functions. *Systems Engineering, International Conference on*, 0:55–58, 2008. doi: <http://doi.ieeecomputersociety.org/10.1109/ICSEng.2008.77>. (page 45)
- Daniel M Steinberg and Edwin V Bonilla. Extended and Unscented Gaussian Processes. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 1251–1259. Curran Associates, Inc., 2014. URL <http://papers.nips.cc/paper/5455-extended-and-unscented-gaussian-processes.pdf>. (pages 4, 60, and 62)
- Linda S. L. Tan, Victor M. H. Ong, David J. Nott, and Ajay Jasra. Variational inference for sparse spectrum Gaussian process regression. *Statistics and Computing*, 26(6): 1243–1261, 2016. ISSN 1573-1375. doi: 10.1007/s11222-015-9600-7. URL <http://dx.doi.org/10.1007/s11222-015-9600-7>. (page 75)
- Yee Whye Teh. Stick-breaking construction for the Indian buffet process. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, page 2007, 2007. (page 73)
- Michalis K. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *Artificial Intelligence and Statistics 12*, pages 567–574, 2009. (pages 25 and 26)
- Michalis K. Titsias and Miguel Lázaro-Gredilla. Spike and Slab Variational Inference for Multi-Task and Multiple Kernel Learning. In *NIPS 2011*, pages 2339–2347, 2011. (pages 30 and 32)
- Felipe Tobar, Thang D Bui, and Richard E Turner. Learning Stationary Time Series using Gaussian Processes with Nonparametric Kernels. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 3501–3509. Curran Associates, Inc., 2015. (page 50)
- Benjamin P. Tu, Andrzej Kudlicki, Maga Rowicka, and Steven L. McKnight. Logic of the yeast metabolic cycle: Temporal compartmentalization of cellular processes. *Science*, 310(5751):1152–1158, 2005. ISSN 0036-8075. doi: 10.1126/science.1120499. URL <http://science.sciencemag.org/content/310/5751/1152>. (page 36)
- Bo Wahlberg. System Identification Using Laguerre Models. *IEEE Transactions on Automatic Control*, 36(5):551–562, 1991. (page 45)
- Liuping Wang. *Model Predictive Control System Design and Implementation Using MATLAB*. Springer Publishing Company, Incorporated, 1st edition, 2009. ISBN 1848823304, 9781848823303. (page 8)

- Jing Zhao and Shiliang Sun. Variational Dependent Multi-output Gaussian Process Dynamical Systems. In Saso Dzeroski, Pance Panov, Dragi Kocev, and Ljupco Todorovski, editors, *Discovery Science - 17th International Conference, DS 2014, Bled, Slovenia, October 8-10, 2014. Proceedings*, volume 8777 of *Lecture Notes in Computer Science*, pages 350–361. Springer, 2014. ISBN 978-3-319-11811-6. doi: 10.1007/978-3-319-11812-3_30. URL https://doi.org/10.1007/978-3-319-11812-3_30.
(page 15)